```
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRRRRRRRRRRR           III      VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD          DDD RRRRRRRRRRRR           III      VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD          DDD RRRRRRRRRRRR           III      VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD          DDD RRR   RRR              III      VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR   RRR              III      VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR   RRR              III      VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR       RRR          III        VVV     VVV    EEE              RRR       RRR
DDD          DDD RRR       RRR          III        VVV     VVV    EEE              RRR       RRR
DDD          DDD RRR       RRR          III        VVV     VVV    EEE              RRR       RRR
DDDDDDDDDDDD     RRR          RRR    IIIIIIIII       VVV         EEEEEEEEEEEEEEE  RRR          RRR
DDDDDDDDDDDD     RRR          RRR    IIIIIIIII       VVV         EEEEEEEEEEEEEEE  RRR          RRR
DDDDDDDDDDDD     RRR          RRR    IIIIIIIII       VVV         EEEEEEEEEEEEEEE  RRR          RRR
```

```
  CCCCCCCC  VV      VV  DDDDDDDD   RRRRRRRR     IIIIII   VV      VV  EEEEEEEEEE  RRRRRRRR
  CCCCCCCC  VV      VV  DDDDDDDD   RRRRRRRR     IIIIII   VV      VV  EEEEEEEEEE  RRRRRRRR
CC          VV      VV  DD      DD RR      RR      II    VV      VV  EE          RR      RR
CC          VV      VV  DD      DD RR      RR      II    VV      VV  EE          RR      RR
CC          VV      VV  DD      DD RR      RR      II    VV      VV  EE          RR      RR
CC          VV      VV  DD      DD RR      RR      II    VV      VV  EE          RR      RR
CC          VV      VV  DD      DD RRRRRRRR      II    VV      VV  EEEEEEEE    RRRRRRRR
CC          VV      VV  DD      DD RRRRRRRR      II    VV      VV  EEEEEEEE    RRRRRRRR
CC          VV      VV  DD      DD RR  RR        II    VV      VV  EE          RR  RR
CC          VV      VV  DD      DD RR  RR        II    VV      VV  EE          RR  RR
CC            VV  VV    DD      DD RR    RR      II      VV  VV    EE          RR    RR      ....
CC            VV  VV    DD      DD RR    RR      II      VV  VV    EE          RR    RR      ....
  CCCCCCCC      VV      DDDDDDDD   RR      RR  IIIIII      VV      EEEEEEEEEE  RR      RR    ....
  CCCCCCCC      VV      DDDDDDDD   RR      RR  IIIIII      VV      EEEEEEEEEE  RR      RR    ....

LL            IIIIII     SSSSSSSS
LL            IIIIII     SSSSSSSS
LL              II     SS
LL              II     SS
LL              II     SS
LL              II     SS
LL              II       SSSSSS
LL              II       SSSSSS
LL              II             SS
LL              II             SS
LL              II             SS
LL              II             SS
LLLLLLLLLL    IIIIII   SSSSSSSS
LLLLLLLLLL    IIIIII   SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  CVDRIVER - VAX/VMS VAX 8600 CONSOLE DISK DRIVER
0000     2          .IDENT  'V04-001'
0000     3  ;
0000     4  ;*****************************************************************************
0000     5  ;*                                                                           *
0000     6  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8  ;*  ALL RIGHTS RESERVED.                                                     *
0000     9  ;*                                                                           *
0000    10  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    11  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    12  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    13  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    14  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    15  ;*  TRANSFERRED.                                                             *
0000    16  ;*                                                                           *
0000    17  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    18  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    19  ;*  CORPORATION.                                                             *
0000    20  ;*                                                                           *
0000    21  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    22  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23  ;*                                                                           *
0000    24  ;*                                                                           *
0000    25  ;*****************************************************************************
0000    26  ;
0000    27  ; FACILITY:
0000    28  ;
0000    29  ;       VAX/VMS VAX 8600 CONSOLE RL02 DRIVER
0000    30  ;
0000    31  ; AUTHOR:
0000    32  ;
0000    33  ;       BENN SCHREIBER, 15-MAR-1983
0000    34  ;
0000    35  ; MODIFIED BY:
0000    36  ;
0000    37  ;       V04-001 BLS0345         Benn Schreiber          27-AUG-1984
0000    38  ;               Retry complete transfer rather than attempting restart
0000    39  ;               at last block.  This avoids forking per-block in the
0000    40  ;               non-error case.  Wait for ready before issuing sts/reset
0000    41  ;               on error path.  Increase timeout on read/write operations.
0000    42  ;               Check for errors on get status interrupts following read/write.
0000    43  ;
0000    44  ;       V03-005 BLS0342         Benn Schreiber          19-AUG-1984
0000    45  ;               Implement abort, reset with status.  Modify cvc_getsts
0000    46  ;               to use TIMEDWAIT macro.
0000    47  ;
0000    48  ;       V03-004 TCM0002         Trudy C. Matthews       09-Aug=1984
0000    49  ;               Increase timeout value in CVC_GETSTS from 30 to 100000.
0000    50  ;
0000    51  ;       V03-003 TCM0001         Trudy C. Matthews       08-Aug-1984
0000    52  ;               Several bug fixes.  Also a spec change - the LBN in STXCS
0000    53  ;               must always be valid for each word of the transfer (and not
0000    54  ;               just the first).
0000    55  ;
0000    56  ;--
```

```
                       ;  ABSTRACT:
                       ;
                       ;       THIS MODULE CONTAINS THE TABLES AND ROUTINES NECESSARY TO
                       ;       PERFORM ALL DEVICE-DEPENDENT PROCESSING OF AN I/O REQUEST
                       ;       FOR RL02 DISK TYPES ON A VAX/VMS VAX 8600 CONSOLE SUBSYSTEM.
                       ;
                       ;       THE DISKS HAVE THE FOLLOWING PHYSICAL GEOMETRY:
                       ;
                       ;                          TRACKS/     SECTORS/     BYTES/      MAXIMUM
                       ;             # CYL        CYLINDER    TRACK        SECTOR      BLOCKS
                       ;
                       ;  RL02        512           2           40           256        20480
                       ;
                       ;       THE IO$X_INHSEEK FUNCTION MODIFIER IS TREATED AS A NO-OP BY
                       ;       THIS DRIVER, SINCE AN IMPLICIT SEEK IS ALWAYS DONE BY THE
                       ;       CONSOLE SUBSYSTEM WHEN READING/WRITING.
                       ;
                       ;--
```

```
                                                    K 4
                     0000     77          .SBTTL  EXTERNAL AND LOCAL DEFINITIONS
                     0000     78
                     0000     79 ;
                     0000     80 ; EXTERNAL SYMBOLS
                     0000     81 ;
                     0000     82
                     0000     83          $CRBDEF                         ;DEFINE CHANNEL REQUEST BLOCK
                     0000     84          $DCDEF                          ;DEFINE DEVICE CLASS
                     0000     85          $DDBDEF                         ;DEFINE DEVICE DATA BLOCK
                     0000     86          $DEVDEF                         ;DEFINE DEVICE CHARACTERISTICS
                     0000     87          $DPTDEF                         ;DEFINE DRIVER PROLOGUE TABLE
                     0000     88          $DYNDEF                         ;DEFINE DYNAMIC DATA STRUCTURE TYPES
                     0000     89          $EMBDEF                         ;DEFINE ERROR MESSAGE BUFFER
                     0000     90          $IDBDEF                         ;DEFINE INTERRUPT DATA BLOCK
                     0000     91          $IODEF                          ;DEFINE I/O FUNCTION CODES
                     0000     92          $IRPDEF                         ;DEFINE I/O REQUEST PACKET
                     0000     93          $PRDEF                          ;DEFINE PROCESSOR REGISTERS
                     0000     94          $SSDEF                          ;DEFINE SYSTEM STATUS CODES
                     0000     95          $UCBDEF                         ;DEFINE UNIT CONTROL BLOCK
                     0000     96          $VECDEF                         ;DEFINE INTERRUPT VECTOR BLOCK
                     0000     97
                     0000     98 ;        GENF
                     0000     99 ;        GENERATE CASE TABLE INDEX SYMBOL
                     0000    100
                     0000    101          .MACRO  GENF FCODE
                     0000    102                  CD'FCODE=$$GENF_CODE
                     0000    103                  $$GENF_CODE=$$GENF_CODE+1
                     0000    104          .ENDM
                     0000    105 ;
                     0000    106 ; LOCAL SYMBOLS
                     0000    107 ;
00000004             0000    108          CV_NUM_REGS     = 4             ;NUMBER OF DEVICE REGISTERS (MIMIC RL02)
00000005             0000    109          CV_SLM          = 5             ;STATE=SEEK LINEAR MODE (READY TO GO)
                     0000    110 ;
                     0000    111 ; UCB OFFSETS WHICH FOLLOW THE STANDARD UCB FIELDS
                     0000    112 ;
                     0000    113          $DEFINI UCB                     ;START OF UCB DEFINITIONS
                     0000    114
000000CC             0000    115 .=UCB$K_LCL_DISK_LENGTH                  ;BEGIN DEFINITIONS AT END OF UCB
                     00CC    116 $DEF     UCB$L_CV_CS        .BLKL   1    ;CONTROL STATUS REGISTER
                     00D0    117 $DEF     UCB$L_CV_MP        .BLKL   1    ;MULTIPURPOSE REGISTER
                     00D4    118 $DEF     UCB$Q_CV_CSMP      .BLKQ   1    ;SAVE CS AND MP DURING RESET STATUS
                     00DC    119 $DEF     UCB$B_CV_STATE     .BLKB   1    ;CURRENT INTERRUPT STATE **ADJACENCY
                     00DD    120 $DEF     UCB$B_CV_STS       .BLKB   1    ;STATUS FLAGS                **ASSUMED
                     00DE    121 $DEF     UCB$W_CV_BBC       .BLKW   1    ;BLOCK BYTE COUNT REMAINING
                     00E0    122 $DEF     UCB$L_CV_IBUF      .BLKL   1    ;INTERNAL BUFFER FOR READING
                     00E4    123 $DEF     UCB$L_CV_MVRTN     .BLKL   1    ;ADDRESS OF BUFFER MOVE ROUTINE
                     00E8    124 $DEF     UCB$L_CV_BUFWIN    .BLKL   1    ;BUFFER WINDOW
                     00EC    125 $DEF     UCB$Q_CV_BDAT      .BLKQ   1    ;SAVPTE AND TRANSFER PARAMS THIS BLOCK
                     00F4    126 $DEF     UCB$L_CV_LBN       .BLKL   1    ;SAVE STARTING LBN OF TRANSFER
                     00F8    127 $DEF     UCB$L_CV_ABPC      .BLKL   1    ;SAVE RETURN ADDRESS FROM ABORT CALL
                     00FC    128 $DEF     UCB$K_CV_LEN       .BLKW   0    ;LENGTH OF UCB
                     00FC    129          $DEFEND UCB                     ;END OF UCB DEFINITONS
                     0000    130
                     0000    131 ;
                     0000    132 ; RL11/RL01 REGISTER OFFSETS FROM CSR ADDRESS
                     0000    133 ;
```

```
                                                L 4
0000   134          $DEFINI CV                          ; START OF REGISTER DEFINITIONS
0000   135
0000   136 ;
0000   137 ; UCB$B_CV_STS FLAGS
0000   138 ;
0000   139          _VIELD  CV,0,<-
0000   140                  <RD,,M>,-                   ;SET IF READ OPERATION
0000   141                  <STSONLY,,M>,-              ;OPERATION IS GET STATUS ONLY
0000   142                  <STSERROR,,M>,-             ;ERROR FROM CONSOLE ON GETSTS INTERRUPT
0000   143                  <ABORT,,M>>                 ;ABORT CURRENT OPERATION AND RETRY
0000   144
0000   145 $DEF     CV_CS               .BLKW   1       ;CONTROL STATUS REGISTER (CSR)
0002   146          _VIELD  CV_CS,0,<-                  ;START OF CSR BIT DEFINITIONS
0002   147                  <DRDY,,M>,-                 ; DRIVE READY
0002   148                  <,3>,-                      ; FUNCTION CODE
0002   149                  <,2>,-                      ; BUS ADDRESS EXTENSION BITS
0002   150                  <,1>,-                      ; INTERRUPT ENABLE
0002   151                  <,1>,-                      ; CONTROLLER READY
0002   152                  <DS,2>,-                    ; DRIVE SELECT
0002   153                  <OPI,,M>,-                  ; OPERATION INCOMPLETE
0002   154                  <CRC,,M>,-                  ; DATA CRC OR HEADER CRC
0002   155                  <CVT,,M>,-                  ; DATA LATE OR HEADER NOT FOUND
0002   156                  <NXM,,M>,-                  ; NON-EXISTENT MEMORY
0002   157                  <DE,,M>,-                   ; DRIVE ERROR
0002   158                  <CE,,M>-                    ; COMPOSITE ERROR
0002   159          >                                   ;END CSR BIT DEFINITIONS
0002   160
0002   161 $DEF     CV_MP               .BLKW   1       ;MULTIPURPOSE REGISTER (MPR)
0004   162          _VIELD  CV_MP,0,<-                  ;START OF MPR BIT DEFINITIONS
0004   163                  <STA,3>,-                   ; DRIVE STATE
0004   164                  <BH,,M>,-                   ; BRUSH HOME
0004   165                  <HO,,M>,-                   ; HEADS OUT
0004   166                  <CO,,M>,-                   ; COVER OPEN
0004   167                  <HS,,M>,-                   ; HEAD SELECT
0004   168                  <,1>,-                      ; DRIVE TYPE
0004   169                  <DSE,,M>,-                  ; DRIVE SELECT ERROR
0004   170                  <VC,,M>,-                   ; VOLUME CHECK
0004   171                  <WGE,,M>,-                  ; WRITE GATE ERROR
0004   172                  <SPE,,M>,-                  ; SPIN ERROR
0004   173                  <SKTO,,M>,-                 ; SEEK TIME OUT
0004   174                  <WL,,M>,-                   ; WRITE LOCK
0004   175                  <CHE,,M>,-                  ; CURRENT HEAD ERROR
0004   176                  <WDE,,M>-                   ; WRITE DATA ERROR
0004   177          >                                   ;END MPR BIT DEFINITIONS
0004   178
0004   179 ;
0004   180 ; VAX 8600 STXCS FORMAT
0004   181 ;
0004   182          _VIELD  STXCS,0,<-
0004   183                  <FUNC,4>,-                  ;DISK FUNCTION TO PERFORM
0004   184                  <,2>,-                      ;MBZ
0004   185                  <IE,,M>,-                   ;INTERRUPT ENABLE
0004   186                  <RDY,,M>,-                  ;READY
0004   187                  <ADDRS,16>,-                ;DISK LOGICAL BLOCK NUMBER
0004   188                  <STS,8>-                    ;STATUS OF TRANSFER
0004   189                  >
0004   190
```

```
                     0004   191          $DEFEND CV                          ;END RL11/RL01 REGISTER DEFINITIONS
                     0000   192
                     0000   193 ;
                     0000   194 ; VAX 8600 CONSOLE STXCS STATUS CODES
                     0000   195 ;
00000001             0000   196 TRANS_COMPLETE = 1                           ;TRANSACTION COMPLETED
00000002             0000   197 TRANS_CONTINUE = 2                           ;CONTINUE TRANSACTION
00000003             0000   198 TRANS_ABORTED = 3                            ;TRANSACTION ABORTED
00000004             0000   199 RETURNED_STATUS = 4                          ;STATUS RETURNED
00000080             0000   200 HANDSHAKE_ERROR = ^X80                       ;HANDSHAKE ERROR DURING TRANSACTION
00000081             0000   201 HW_ERROR = ^X81                              ;HARDWARE ERROR DURING TRANSACTION
                     0000   202 ;
                     0000   203 ; VAX 8600 CONSOLE STXCS FUNCTION CODES
                     0000   204 ;
00000000             0000   205 NO_OP = 0                                    ;NO OPERATION
00000002             0000   206 STATUS_RESET = 2                            ;READ DEVICE STATUS WITH RST ASSERTED
00000003             0000   207 ABORT_TRANSFER = 3                          ;ABORT CURRENT TRANSFER
00000004             0000   208 READ_STATUS = 4                             ;READ DEVICE STATUS
00000005             0000   209 WRITE_BLOCK = 5                             ;WRITE BLOCK OF DATA
00000006             0000   210 READ_BLOCK = 6                              ;READ BLOCK OF DATA
                     0000   211 ;
                     0000   212 ; INTERRUPT TRANSITION CODES
                     0000   213 ;
00000000             0000   214 ITC_DATA = 0                                ;READ OR WRITE DATA
00000001             0000   215 ITC_STS1 = 1                                ;GET CONTROL/STATUS REGISTER
00000002             0000   216 ITC_STS2 = 2                                ;GET RL11 MULTIPURPOSE REGISTER
00000003             0000   217 ITC_ABORT = 3                               ;ABORT CURRENT TRANSFER
00000004             0000   218 ITC_RESET1 = 4                              ;GET CONTROL/STATUS WITH RST ASSERTED
00000005             0000   219 ITC_RESET2 = 5                              ;GET MP REG WITH RST ASSERTED
                     0000   220 ;
                     0000   221 ; DEFINE THE CASE OFFSETS AS CD'FUNCTION (I.E. CDF_NOP)
                     0000   222 ;
00000000             0000   223          $$GENF_CODE = 0                    ;INITIALIZE
                     0000   224          GENF    F_NOP                      ;NO-OP
                     0000   225          GENF    F_UNLOAD                   ;UNLOAD VOLUME (NOP)
                     0000   226          GENF    F_SEEK                     ;SEEK
                     0000   227          GENF    F_RECAL                    ;RECALIBRATE (NOP)
                     0000   228          GENF    F_DRVCLR                   ;DRIVE CLEAR (RESET & GET STATUS)
                     0000   229          GENF    F_RELEASE                  ;RELEASE PORT (NOP)
                     0000   230          GENF    F_OFFSET                   ;OFFSET HEADS (NOP)
                     0000   231          GENF    F_RETCENTER                ;RETURN HEADS TO CENTERLINE (NOP)
                     0000   232          GENF    F_PACKACK                  ;PACK ACKNOWLEDGE (RESET & GET STATUS)
                     0000   233          GENF    F_SEARCH                   ;SEARCH (NOP)
                     0000   234          GENF    F_WRITECHECK               ;WRITE CHECK
                     0000   235          GENF    F_WRITEDATA                ;WRITE DATA
                     0000   236          GENF    F_READDATA                 ;READ DATA
                     0000   237          GENF    F_WRITEHEAD                ;WRITE HEADERS (NOP)
                     0000   238          GENF    F_READHEAD                 ;READ HEADERS
                     0000   239          GENF    F_NOP                      ;place holder
                     0000   240          GENF    F_NOP                      ;place holder
                     0000   241          GENF    F_AVAILABLE                ;AVAILABLE
                     0000   242
```

```
                        .SBTTL  STANDARD TABLES
                ;
                ; DRIVER PROLOGUE TABLE
                ;
                ;       THE DPT DESCRIBES DRIVER PARAMETERS AND I/O DATABASE FIELDS
                ;       THAT ARE TO BE INITIALIZED DURING DRIVER LOADING AND RELOADING
                ;

                        DPTAB   -                               ;DPT CREATION MACRO
                                END=CV_END,-                    ;END OF DRIVER LABEL
                                ADAPTER=UBA,-                   ;ADAPTER TYPE = UNIBUS
                                FLAGS=DPT$M_SVP,-               ;SYSTEM PAGE TABLE ENTRY REQUIRED
                                UCBSIZE=UCB$K_CV_LEN,-          ;LENGTH OF UCB
                                NAME=CVDRIVER                   ;DRIVER NAME

                        DPT_STORE INIT                          ;START CONTROL BLOCK INIT VALUES
                        DPT_STORE DDB,DDB$L_ACPD,L,<^A\F11\>  ;DEFAULT ACP NAME
                        DPT_STORE DDB,DDB$L_ACPD+3,B,DDB$K_CART ;ACP CLASS
                        DPT_STORE UCB,UCB$B_FIPL,B,8            ;FORK IPL
                        DPT_STORE UCB,UCB$L_DEVCHAR,L,-         ;DEVICE CHARACTERISTICS
                                <DEV$M_FOD-                     ; FILES ORIENTED
                                !DEV$M_DIR-                     ; DIRECTORY STRUCTURED
                                !DEV$M_AVL-                     ; AVAILABLE
                                !DEV$M_ELG-                     ; ERROR LOGGING
                                !DEV$M_SHR-                     ; SHAREABLE
                                !DEV$M_IDV-                     ; INPUT DEVICE
                                !DEV$M_ODV-                     ; OUTPUT DEVICE
                                !DEV$M_RND>                     ; RANDOM ACCESS
                        DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_DISK  ;DEVICE CLASS
                        DPT_STORE UCB,UCB$W_DEVBUFSIZ,W,512  ;DEFAULT BUFFER SIZE
                        DPT_STORE UCB,UCB$B_SECTORS,B,40  ;NUMBER OF SECTORS PER TRACK
                        DPT_STORE UCB,UCB$B_TRACKS,B,2   ;NUMBER OF TRACKS PER CYLINDER
                        DPT_STORE UCB,UCB$B_DIPL,B,21    ;DEVICE IPL
                        DPT_STORE UCB,UCB$B_ERTMAX,B,8   ;MAX ERROR RETRY COUNT

                        DPT_STORE REINIT                        ;START CONTROL BLOCK RE-INIT VALUES
                        DPT_STORE CRB,CRB$L_INTD+4,D,CV_INT  ;INTERRUPT SERVICE ROUTINE ADDRESS
                        DPT_STORE CRB,CRB$L_INTD+VEC$L_INITIAL,-  ;CONTROLLER INIT ADDRESS
                                D,CV_RL11_INIT                  ;...
                        DPT_STORE CRB,CRB$L_INTD+VEC$L_UNITINIT,- ;UNIT INIT ADDRESS
                                D,CV_RLOX_INIT                  ;...
                        DPT_STORE DDB,DDB$L_DDT,D,CV$DDT        ;DDT ADDRESS

                        DPT_STORE END                   ;END OF INITIALIZATION TABLE

                ;
                ; DRIVER DISPATCH TABLE
                ;
                ;       THE DDT LISTS ENTRY POINTS FOR DRIVER SUBROUTINES WHICH ARE
                ;       CALLED BY THE OPERATING SYSTEM.
                ;

                        DDTAB   -                               ;DDT CREATION MACRO
                                DEVNAM=CV,-                     ;NAME OF DEVICE
                                START=CV_STARTIO,-              ;START I/O ROUTINE
                                UNSOLIC=CV_UNSOLNT,-            ;UNSOLICITED INTERRUPT
```

```
                        ;
                        ; STANDARD TABLES
                        ;
                        ;+
                        ; FUNCTION DECISION TABLE
                        ;
                        ; FORMAT:
                        ;
                        ;       FUNCTB=CV_FUNCTABLE,-
                        ;       CANCEL=0,-
                        ;       REGDMP=CV_REGDUMP,-
                        ;       DIAGBF=<<CV_NUM_REGS+5+5+3+1>*4>,-
                        ;       ERLGBF=<<<CV_NUM_REGS+5+1>*4>+EMB$L_DV_REGSAV>
                        ;
                        ; ARGUMENTS:
                        ;
                        ;       FUNCTB  ADDRESS OF FUNCTION DECISION TABLE
                        ;       CANCEL  ADDRESS OF CANCEL ROUTINE
                        ;       REGDMP  ADDRESS OF REGISTER DUMP ROUTINE
                        ;       DIAGBF  BYTES IN DIAGNOSTIC BUFFER
                        ;       ERLGBF  BYTES IN ERROR LOG BUFFER
                        ;
                        ; DIAGNOSTIC BUFFER SIZE = <<4 RL02 REGISTER LONGWORDS + 5 UCB FIELD LONGWORDS
                        ;                         + 5 IOC$DIAGBUFILL LONGWORDS + 3 BUFFER ALLOCATION
                        ;                         LONGWORDS + 1 LONGWORD FOR # REGISTERS IN CV_REGDUMP>
                        ;                         * 4 BYTES/LONGWORD>
                        ;
                        ; ERROR LOG BUFFER SIZE  = <<<4 RL02 REGISTER LONGWORDS + 5 UCB FIELD LONGWORDS
                        ;                         + 1 LONGWORD FOR # REGISTERS IN CV_REGDUMP>
                        ;                         * 4 BYTES/LONGWORD> + BYTES NEEDED FOR ERROR LOGGER
                        ;                         TO SAVE SOFTWARE REGISTERS>
                        ;-
```

```
                        ;
                        ; FUNCTION DECISION TABLE
                        ;
                        ;       THE FDT LISTS VALID FUNCTION CODES, SPECIFIES WHICH
                        ;       CODES ARE BUFFERED, AND DESIGNATES SUBROUTINES TO
                        ;       PERFORM PREPROCESSING FOR PARTICULAR FUNCTIONS.
                        ;
                        CV_FUNCTABLE:
                                FUNCTAB ,-                      ;LIST LEGAL FUNCTIONS
                                        <NOP,-                  ; NO-OP
                                        UNLOAD,-                ; UNLOAD
                                        SEEK,-                  ; SEEK
                                        DRVCLR,-                ; DRIVE CLEAR
                                        PACKACK,-               ; PACK ACKNOWLEDGE
                                        SENSECHAR,-             ; SENSE CHARACTERISTICS
                                        SETCHAR,-               ; SET CHARACTERISTICS
                                        SENSEMODE,-             ; SENSE MODE
                                        SETMODE,-               ; SET MODE
                                        READLBLK,-              ; READ LOGICAL BLOCK
                                        WRITELBLK,-             ; WRITE LOGICAL BLOCK
                                        READPBLK,-              ; READ PHYSICAL BLOCK
                                        WRITEPBLK,-             ; WRITE PHYSICAL BLOCK
                                        READVBLK,-              ; READ VIRTUAL BLOCK
                                        WRITEVBLK,-             ; WRITE VIRTUAL BLOCK
                                        AVAILABLE,-             ; AVAILABLE
                                        ACCESS,-                ; ACCESS FILE / FIND DIRECTORY ENTRY
                                        ACPCONTROL,-            ; ACP CONTROL FUNCTION
                                        CREATE,-                ; CREATE FILE AND/OR DIRECTORY ENTRY
                                        DEACCESS,-              ; DEACCESS FILE
                                        DELETE,-                ; DELETE FILE AND/OR DIRECTORY ENTRY
                                        MODIFY,-                ; MODIFY FILE ATTRIBUTES
                                        MOUNT-                  ; MOUNT VOLUME
                                        >
                                FUNCTAB ,-                      ;BUFFERED FUNCTIONS
                                        <NOP,-                  ; NO-OP
                                        UNLOAD,-                ; UNLOAD
                                        SEEK,-                  ; SEEK
                                        DRVCLR,-                ; DRIVE CLEAR
                                        PACKACK,-               ; PACK ACKNOWLEDGE
                                        SENSECHAR,-             ; SENSE CHARACTERISTICS
                                        SETCHAR,-               ; SET CHARACTERISTICS
                                        SENSEMODE,-             ; SENSE MODE
                                        SETMODE,-               ; SET MODE
                                        AVAILABLE,-             ; AVAILABLE
                                        ACCESS,-                ; ACCESS FILE / FIND DIRECTORY ENTRY
                                        ACPCONTROL,-            ; ACP CONTROL FUNCTION
                                        CREATE,-                ; CREATE FILE AND/OR DIRECTORY ENTRY
                                        DEACCESS,-              ; DEACCESS FILE
                                        DELETE,-                ; DELETE FILE AND/OR DIRECTORY ENTRY
                                        MODIFY,-                ; MODIFY FILE ATTRIBUTES
                                        MOUNT-                  ; MOUNT VOLUME
                                        >
                                FUNCTAB CV_ALIGN,-              ;TEST ALIGNMENT FUNCTIONS
                                        <READLBLK,-             ; READ LOGICAL BLOCK
                                        READPBLK,-              ; READ PHYSICAL BLOCK
                                        READVBLK,-              ; READ VIRTUAL BLOCK
```

```
                                                D 5
0048   377                      WRITELBLK,-             ; WRITE LOGICAL BLOCK
0048   378                      WRITEPBLK,-             ; WRITE PHYSICAL BLOCK
0048   379                      WRITEVBLK-              ; WRITE VIRTUAL BLOCK
0048   380                      >
0054   381              FUNCTAB +ACP$READBLK,-          ;READ FUNCTIONS
0054   382                      <READLBLK,-             ; READ LOGICAL BLOCK
0054   383                      READPBLK,-              ; READ PHYSICAL BLOCK
0054   384                      READVBLK-               ; READ VIRTUAL BLOCK
0054   385                      >
0060   386              FUNCTAB +ACP$WRITEBLK,-         ;WRITE FUNCTIONS
0060   387                      <WRITELBLK,-            ; WRITE LOGICAL BLOCK
0060   388                      WRITEPBLK,-             ; WRITE PHYSICAL BLOCK
0060   389                      WRITEVBLK-              ; WRITE VIRTUAL BLOCK
0060   390                      >
006C   391              FUNCTAB +ACP$ACCESS,-           ;ACCESS FUNCTIONS
006C   392                      <ACCESS,-               ; ACCEESS FILE / FIND DIRECTORY ENTRY
006C   393                      CREATE-                 ; CREATE FILE AND/OR DIRECTORY ENTRY
006C   394                      >
0078   395              FUNCTAB +ACP$DEACCESS,-         ;DEACCESS FUNCTION
0078   396                      <DEACCESS-              ; DEACCESS FILE
0078   397                      >
0084   398              FUNCTAB +ACP$MODIFY,-           ;MODIFY FUNCTIONS
0084   399                      <ACPCONTROL,-           ; ACP CONTROL FUNCTION
0084   400                      DELETE,-                ; DELETE FILE AND/OR DIRECTORY ENTRY
0084   401                      MODIFY-                 ; MODIFY FILE ATTRIBUTES
0084   402                      >
0090   403              FUNCTAB +ACP$MOUNT,-            ;MOUNT FUNCTION
0090   404                      <MOUNT-                 ; MOUNT VOLUME
0090   405                      >
009C   406              FUNCTAB +EXE$LCLDSKVALID,-
009C   407                      <UNLOAD,-
009C   408                      AVAILABLE,-
009C   409                      PACKACK-
009C   410                      >
00A8   411              FUNCTAB +EXE$ZEROPARM,-         ;ZERO PARAMETER FUNCTIONS
00A8   412                      <NOP,-                  ; NO-OP
00A8   413                      UNLOAD,-                ; UNLOAD
00A8   414                      DRVCLR,-                ; DRIVE CLEAR
00A8   415                      PACKACK,-               ; PACK ACKNOWLEDGE
00A8   416                      AVAILABLE,-             ; AVAILABLE
00A8   417                      >
00B4   418              FUNCTAB +EXE$ONEPARM,-          ;ONE PARAMETER FUNCTION
00B4   419                      <SEEK-                  ; SEEK
00B4   420                      >
00C0   421              FUNCTAB +EXE$SENSEMODE,-        ;SENSE FUNCTIONS
00C0   422                      <SENSECHAR,-            ; SENSE CHARACTERISTICS
00C0   423                      SENSEMODE-              ; SENSE MODE
00C0   424                      >
00CC   425              FUNCTAB +EXE$SETCHAR,-          ;SET FUNCTIONS
00CC   426                      <SETCHAR,-              ; SET CHARACTERISTICS
00CC   427                      SETMODE-                ; SET MODE
00CC   428                      >
```

```
                        00D8    430
                        00D8    431                 .SBTTL  CONTROLLER INITIALIZATION ROUTINE
                        00D8    432 ;++
                        00D8    433 ;
                        00D8    434 ; FUNCTIONAL DESCRIPTION:
                        00D8    435 ;
                        00D8    436 ;       THIS ROUTINE IS A NO-OP FOR THE RL11 BUT MUST BE INCLUDED
                        00D8    437 ;       SINCE IT IS CALLED WHEN THE RL02 IS BOOTED AS A SYSTEM DEVICE.
                        00D8    438 ;
                        00D8    439 ;       THE OPERATING SYSTEM CALLS THIS ROUTINE:
                        00D8    440 ;               - AT SYSTEM STARTUP
                        00D8    441 ;               - DURING DRIVER LOADING
                        00D8    442 ;               - DURING RECOVERY FROM POWER FAILURE
                        00D8    443 ;
                        00D8    444 ; INPUTS:
                        00D8    445 ;
                        00D8    446 ;       R4      - CSR ADDRESS (DEVICE CONTROL STATUS REGISTER)
                        00D8    447 ;       R5      - IDB ADDRESS (INTERRUPT DATA BLOCK)
                        00D8    448 ;       ALL INTERRUPTS ARE LOCKED OUT
                        00D8    449 ;
                        00D8    450 ; OUTPUTS:
                        00D8    451 ;
                        00D8    452 ;       CONTROL IS RETURNED TO THE CALLER.
                        00D8    453 ;
                        00D8    454 ;--
                        00D8    455
                        00D8    456 CV_RL11_INIT:                               ;CONTROLLER INITIALIZATION
                    05  00D8    457         RSB                                 ;RETURN TO CALLER
```

```
                        00D9   459          .SBTTL  UNIT INITIALIZATION ROUTINE
                        00D9   460
                        00D9   461 ;++
                        00D9   462 ;
                        00D9   463 ; CV_RL0x_INIT - UNIT INITIALIZATION ROUTINE
                        00D9   464 ;
                        00D9   465 ; FUNCTIONAL DESCRIPTION:
                        00D9   466 ;
                        00D9   467 ;       THIS ROUTINE READIES THE RL02 UNIT FOR I/O OPERATIONS.
                        00D9   468 ;
                        00D9   469 ;       THE OPERATING SYSTEM CALLS THIS ROUTINE:
                        00D9   470 ;               - AT SYSTEM STARTUP
                        00D9   471 ;               - DURING DRIVER LOADING
                        00D9   472 ;               - DURING RECOVERY FROM POWER FAILURE
                        00D9   473 ;
                        00D9   474 ; INPUTS:
                        00D9   475 ;
                        00D9   476 ;       R4      - CSR ADDRESS (CONTROLLER STATUS REGISTER)
                        00D9   477 ;       R5      - UCB ADDRESS (UNIT CONTROL BLOCK)
                        00D9   478 ;
                        00D9   479 ; OUTPUTS:
                        00D9   480 ;
                        00D9   481 ;       THE DRIVE UNIT IS RESET, UCB FIELDS ARE INITIALIZED, AND THE
                        00D9   482 ;       ROUTINE WAITS FOR ONLINE UNITS TO SPIN UP.  ALL REGISTERS
                        00D9   483 ;       EXCEPT R0-R3 ARE PRESERVED.
                        00D9   484 ;
                        00D9   485 ;--
                        00D9   486
                        00D9   487 CV_RL0X_INIT:                            ;RL01/RL02 UNIT INITIALIZATION
          0810 8F   AA  00D9   488          BICW2   #<UCB$M_ONLINE!UCB$M_VALID>,-  ;ASSUME OFFLINE/INVALID
            64 A5       00DD   489                  UCB$W_STS(R5)             ;...
                        00DF   490
      2324C002 8F   D0  00DF   491          MOVL    #^X2324C002,-             ;SET MEDIA IDENT "DL RL02"
            008C C5     00E5   492                  UCB$L_MEDIA_ID(R5)
              0A 90     00E8   493          MOVB    S^#DT$_RL02,-             ;SET RL02 DEVICE TYPE
              41 A5     00EA   494                  UCB$B_DEVTYPE(R5)
  46 A5     0200 8F   B0  00EC   495          MOVW    #512,UCB$W_CYLINDERS(R5);SET NUMBER OF RL02 CYLINDERS
00B0 C5     5000 8F   3C  00F2   496          MOVZWL  #20480,UCB$L_MAXBLOCK(R5);SET MAX RL02 BLOCK NUMBER
              0F 10     00F9   497          BSBB    CVC_GETSTS                ;GET CONSOLE RL02 STATUS
        06 50   00 E1   00FB   498          BBC     #CV_CS_V_DRDY,R0,40$      ;BRANCH IF DRIVE NOT READY
  64 A5     0800 8F   A8  00FF   499          BISW2   #UCB$M_VALID,UCB$W_STS(R5) ;YES, SET VOLUME VALID
        64 A5   10  A8  0105   500 40$:      BISW2   #UCB$M_ONLINE,UCB$W_STS(R5);SET UNIT ONLINE
                    05  0109   501 60$:      RSB
```

```
                                010A   503          .SBTTL  DRIVER SPECIFIC SUBROUTINES
                                010A   504 ;
                                010A   505 ; CVC_GETSTS - GET STATUS FOR VAX 8600 CONSOLE RL02 WITHOUT INTERRUPTS
                                010A   506 ;
                                010A   507 ; INPUTS:
                                010A   508 ;
                                010A   509 ;       NONE
                                010A   510 ;
                                010A   511 ; OUTPUTS:
                                010A   512 ;
                                010A   513 ;       R0 = 0 IF FAILED TO GET STATUS
                                010A   514 ;          = RL02 CONTROL STATUS REGISTER
                                010A   515 ;
                                010A   516 ;       R1 = RL02 MULTIPURPOSE REGISTER (UNUSABLE IF R0=0)
                                010A   517 ;
                                010A   518 CVC_GETSTS:
                          52    DD    010A   519          PUSHL   R2                      ;SAVE R2
                          0D    10    010C   520          BSBB    100$                    ;READ CONTROL STATUS REGISTER
                          50    DD    010E   521          PUSHL   R0                      ;SAVE R0
                          09    10    0110   522          BSBB    100$                    ;READ MULTIPURPOSE REGISTER
                    51    50    D0    0112   523          MOVL    R0,R1                   ;POSITION MULTIPURPOSE REGISTER
                          50  8ED0    0115   524          POPL    R0                      ;RESTORE CSR
                          04    BA    0118   525          POPR    #^M<R2>                 ;RESTORE R2
                                05    011A   526          RSB
                                      011B   527
          0000004C  8F    02    DA    011B   528 100$:    MTPR    #STATUS_RESET,#PR$_STXCS ;REQUEST READ STATUS
                                      0122   529          TIMEDWAIT TIME=#600*1000,-
                                      0122   530                  INS1=<MFPR      #PR$_STXCS,R2>,- ;READ STATUS
                                      0122   531                  INS2=<BBS       #STXCS_V_RDY,R2,140$> ;BRANCH IF READY
                                05    014F   532          RSB                             ;CONSOLE NEVER GOT READY (TIMEDWAIT
                                      0150   533                                          ;CLEARS R0 ON FALL-OUT)
                             5E   04    C0    0150   534 140$:    ADDL2   #4,SP           ;CLEAR TIMEDWAIT'S COUNTER FROM STACK
          50    0000004D  8F    DB    0153   535          MFPR    #PR$_STXDB,R0           ;OBTAIN STATUS FROM CONSOLE
                                05    015A   536          RSB
```

```
                     015B   538          .SBTTL  FDT ROUTINE - TEST TRANSFER BYTE COUNT ALIGNMENT
                     015B   539
                     015B   540 ;++
                     015B   541 ;
                     015B   542 ; DL_ALIGN - FDT ROUTINE TO TEST XFER BYTE COUNT
                     015B   543 ;
                     015B   544 ; FUNCTIONAL DESCRIPTION:
                     015B   545 ;
                     015B   546 ;       THIS ROUTINE IS CALLED FROM THE FUNCTION DECISION TABLE DISPATCHER
                     015B   547 ;       TO CHECK THE BYTE COUNT PARAMETER SPECIFIED BY THE USER PROCESS
                     015B   548 ;       FOR AN EVEN NUMBER OF BYTES (WORD BOUNDARY).
                     015B   549 ;
                     015B   550 ; INPUTS:
                     015B   551 ;
                     015B   552 ;       R3      - IRP ADDRESS (I/O REQUEST PACKET)
                     015B   553 ;       R4      - PCB ADDRESS (PROCESS CONTROL BLOCK)
                     015B   554 ;       R5      - UCB ADDRESS (UNIT CONTROL BLOCK)
                     015B   555 ;       R6      - CCB ADDRESS (CHANNEL CONTROL BLOCK)
                     015B   556 ;       R7      - BIT NUMBER OF THE I/O FUNCTION CODE
                     015B   557 ;       R8      - ADDRESS OF FDT TABLE ENTRY FOR THIS ROUTINE
                     015B   558 ;       4(AP)   - ADDRESS OF FIRST FUNCTION DEPENDENT QIO PARAMETER
                     015B   559 ;
                     015B   560 ; OUTPUTS:
                     015B   561 ;
                     015B   562 ;       IF THE QIO BYTE COUNT PARAMETER IS ODD, THE I/O OPERATION IS
                     015B   563 ;       TERMINATED WITH AN ERROR. IF IT IS EVEN, CONTROL IS RETURNED
                     015B   564 ;       TO THE FDT DISPATCHER.
                     015B   565 ;
                     015B   566 ;--
                     015B   567
                     015B   568 CV_ALIGN:                                    ;CHECK BYTE COUNT AT P1(AP)
 01 04 AC    E8      015B   569          BLBS    4(AP),10$                  ;IF LBS - ODD BYTE COUNT
             05      015F   570          RSB                                ;EVEN - RETURN TO CALLER
50   034C 8F    3C  0160   571 10$:     MOVZWL  #SS$_IVBUFLEN,R0           ;SET BUFFER ALIGNMENT STATUS
  00000000'GF    17  0165   572          JMP     G^EXE$ABORTIO              ;ABORT I/O
```

```
                                                                .SBTTL  START I/O ROUTINE
                                                        ;++
                                                        ;
                                                        ; CV_STARTIO - START I/O ROUTINE
                                                        ;
                                                        ; FUNCTIONAL DESCRIPTION:
                                                        ;
                                                        ;       THIS FORK PROCESS IS ENTERED FROM THE EXECUTIVE AFTER AN I/O REQUEST
                                                        ;       PACKET HAS BEEN DEQUEUED, AND PERFORMS THE FOLLOWING:
                                                        ;
                                                        ;               - ACTIVATES THE CONSOLE AFTER SETTING UCB FIELDS, AND OBTAINING
                                                        ;                 CONTROLLER RESOURCES
                                                        ;
                                                        ;               - WAITS FOR AN INTERRUPT
                                                        ;
                                                        ;               - REGAINS CONTROL AFTER THE ISR SERVICES THE INTERRUPT, AND
                                                        ;                       -  RE-ACTIVATES THE CONSOLE IF THE ORIGINAL FUNCTION
                                                        ;                          IS A RETRIABLE ERROR, OR
                                                        ;                       -  COMPLETES THE I/O REQUEST BY RELEASING RESOURCES,
                                                        ;                          SETTING STATUS CODES, AND RETURNING TO THE EXECUTIVE.
                                                        ;
                                                        ; INPUTS:
                                                        ;
                                                        ;       R3              - IRP ADDRESS (I/O REQUEST PACKET)
                                                        ;       R5              - UCB ADDRESS (UNIT CONTROL BLOCK)
                                                        ;
                                                        ; OUTPUTS:
                                                        ;
                                                        ;       R0      - FIRST I/O STATUS LONGWORD: STATUS CODE & BYTES XFERED
                                                        ;       R1      - SECOND I/O STATUS LONGWORD: 0 FOR DISKS
                                                        ;
                                                        ;       THE I/O FUNCTION IS EXECUTED.
                                                        ;
                                                        ;       ALL REGISTERS EXCEPT R0-R4 ARE PRESERVED.
                                                        ;
                                                        ;--
                                                        CV_STARTIO:                             ;START I/O OPERATION
                                                        ;
                                                        ;       PREPROCESS UCB FIELDS
                                                        ;
                                                        PREPROCESS:
                                                        ;
                                                        ; Convert the physical media address in IRP$L_MEDIA to an LBN.
                                                        ; This is necessary because the console RL02 controller expects an LBN,
                                                        ; not a physical media address.  The LBN is given by the formula:
                                                        ;
                                                        ; LBN = (CYLINDER*(TRACKS/CYLINDER)+TRACK)*(SECTORS/TRACK))+SECTOR
                                                        ;
        38 A3   D0                                      MOVL    IRP$L_MEDIA(R3),-       ;Copy media address to UCB
    00BC C5                                             UCB$L_MEDIA(R5)         ;
53  00BC C5     DE                                      MOVAL   UCB$L_MEDIA(R5),R3      ;Get address of media address
    50   83     9A                                      MOVZBL  (R3)+,R0                ;Get SECTOR
    51   83     9A                                      MOVZBL  (R3)+,R1                ;Get TRACK
    7E   83     3C                                      MOVZWL  (R3)+,-(SP)             ;Get CYLINDER
  53   45 A5     9A                                     MOVZBL  UCB$B_TRACKS(R5),R3     ;Get TRACKS/CYLINDER
```

```
                53    8E   C4   0183   631          MULL2   (SP)+,R3                ;R3 = C*(T/C)
                51    53   C0   0186   632          ADDL2   R3,R1                   ;R1 = C*(T/C)+T
              53    44 A5   9A   0189   633          MOVZBL  UCB$B_SECTORS(R5),R3    ;Get SECTORS/TRACK
                51    53   C4   018D   634          MULL2   R3,R1                   ;R1 = (C*(T/C)+T)*(S/T)
     00BC C5    50    51   C1   0190   635          ADDL3   R1,R0,UCB$L_MEDIA(R5)   ;CALULATE AND STORE LBN
     00F4 C5  00BC C5   D0   0196   636          MOVL    UCB$L_MEDIA(R5),UCB$L_CV_LBN(R5) ;SAVE STARTING LBN FOR RETRIES
              0081 C5   90   019D   637          MOVB    UCB$B_ERTMAX(R5),-      ;INITIALIZE ERROR RETRY COUNT
              0080 C5        01A1   638                  UCB$B_ERTCNT(R5)        ;...
                53    58 A5   D0   01A4   639          MOVL    UCB$L_IRP(R5),R3        ;GET IRP ADDRESS
         009A C5    20 A3   B0   01A8   640          MOVW    IRP$W_FUNC(R3),UCB$W_FUNC(R5)  ;SAVE FUNCTION CODE
                    00   EF   01AE   641          EXTZV   #IRP$V_FCODE,-          ;EXTRACT I/O FUNCTION CODE
              51    20 A3   06   01B0   642                  #IRP$S_FCODE,IRP$W_FUNC(R3),R1 ;...
           0092 C5    51   90   01B4   643          MOVB    R1,UCB$B_FEX(R5)        ;STORE FUNCTION DISPATCH INDEX
                  78 A5   7D   01B9   644          MOVQ    UCB$L_SVAPTE(R5),-      ;SAVE TRANSFER PARAMETERS
                00EC C5        01BC   645                  UCB$Q_CV_BDAT(R5)
                    02   AA   01BF   646          BICW2   #UCB$M_DIAGBUF,-
                  68 A5        01C1   647                  UCB$W_DEVSTS(R5)        ;CLR DIAGNOSTIC BUFFER PRESENT
                    07   E1   01C3   648          BBC     #IRP$V_DIAGBUF,-        ;IF CLR - NO DIAG BUFFER
                04 2A A3        01C5   649                  IRP$W_STS(R3),FDISPATCH ;...
              68 A5    02   A8   01C8   650          BISW2   #UCB$M_DIAGBUF,UCB$W_DEVSTS(R5)  ;SET DIAG BUFFER PRESENT
                               01CC   651 ;
                               01CC   652 ;       CENTRAL FUNCTION DISPATCH
                               01CC   653 ;
                               01CC   654 FDISPATCH:                               ;FUNCTION DISPATCH
                               01CC   655 ;
                               01CC   656 ; RETRY LOGIC IS DONE BY RESTARTING THE ENTIRE TRANSFER, RATHER THAN
                               01CC   657 ; AT THE BLOCK IN ERROR.  HENCE, WE RESTORE TRANSFER PARAMETERS HERE
                               01CC   658 ;
         00C0 C5    7E A5   AE   01CC   659          MNEGW   UCB$W_BCNT(R5),UCB$W_BCR(R5)  ;INIT NEG BYTES LEFT TO XFER
              00EC C5   7D   01D2   660          MOVQ    UCB$Q_CV_BDAT(R5),-     ;RESTORE TRANSFER PARAMETERS
                  78 A5        01D6   661                  UCB$L_SVAPTE(R5)
     00BC C5  00F4 C5   D0   01D8   662          MOVL    UCB$L_CV_LBN(R5),UCB$L_MEDIA(R5) ;RESTORE STARTING LBN
                               01DF   663
                53    58 A5   D0   01DF   664          MOVL    UCB$L_IRP(R5),R3        ;GET IRP ADDRESS
                      08   E0   01E3   665          BBS     #IRP$V_PHYSIO,-         ;IF SET - PHYSICAL I/O FUNCTION
                  0D 2A A3        01E5   666                  IRP$W_STS(R3),10$       ;...
                      0B   E0   01E8   667          BBS     #UCB$V_VALID,-          ;IF SET - VOLUME SOFTWARE VALID
                08 64 A5        01EA   668                  UCB$W_STS(R5),10$       ;...
              50    0254 8F   3C   01ED   669          MOVZWL  #SS$_VOLINV,R0          ;SET VOLUME INVALID STATUS
                    0240   31   01F2   670          BRW     RESETXFR                ;RESET BYTE COUNT AND EXIT
              53    0092 C5   9A   01F5   671 10$:     MOVZBL  UCB$B_FEX(R5),R3        ;GET FUNCTION DISPATCH INDEX
                    00DC C5   B4   01FA   672          CLRW    UCB$B_CV_STATE(R5)      ;CLEAR INTERRUPT STATE AND STATUS
                    00E8 C5   D4   01FE   673          CLRL    UCB$L_CV_BUFWIN(R5)     ;CLEAR BUFFER WINDOW
                               0202   674          CASE    R3,<-                   ;DISPATCH TO FUNCTION HANCVING ROUTINE
                               0202   675                  NOP,-                   ; NOP
                               0202   676                  UNLOAD,-                ; UNLOAD
                               0202   677                  SEEK,-                  ; SEEK
                               0202   678                  NOP,-                   ; RECALIBRATE (unsupported)
                               0202   679                  DRVCLR,-                ; DRVCLR
                               0202   680                  NOP,-                   ; RELEASE PORT (unsupported)
                               0202   681                  NOP,-                   ; OFFSET HEADS (unsupported)
                               0202   682                  NOP,-                   ; RETURN TO CENTER (unsupported)
                               0202   683                  PACKACK,-               ; PACK ACKNOWLEDGE
                               0202   684                  NOP,-                   ; SEARCH (unsupported)
                               0202   685                  WRITECHECK,-            ; WRITE CHECK (unsupported)
                               0202   686                  WRITEDATA,-             ; WRITE DATA
                               0202   687                  READDATA,-              ; READ DATA
```

```
                                                 K 5
                     0202   688                      NOP,-                   ; WRITE HEADER (unsupported)
                     0202   689                      NOP,-                   ; READ HEADER
                     0202   690                      NOP,-                   ; place holder
                     0202   691                      NOP,-                   ; place holder
                     0202   692                      AVAILABLE-              ; AVAILABLE
                     0202   693                      >
                     022A   694
                     022A   695 NOP:                                          ;NO-OP
                     022A   696 WRITECHECK:                                   ;WRITE CHECK
                     022A   697 SEEK:                                         ;SEEK
                     022A   698 DRVCLR:                                       ;DRIVE CLEAR (GET STATUS & RESET)
                     022A   699 DO_FUNCTION:
          0092 C5    94   022A   700          CLRB    UCB$B_FEX(R5)           ;SET FUNCTION
               0097  30   022E   701          BSBW    FEXL                    ;EXECUTE FUNCTION
                     20'  0231   702          .BYTE   RETRYERR-.-1            ;ERROR OFFSET
                     11   0232   703          BRB     NORMAL                  ;DONE
                     0234   704
                     0234   705 PACKACK:                                      ;PACK ACKNOWLEDGE (GET STATUS & RESET)
64 A5     0800 8F    A8   0234   706          BISW2   #UCB$M_VALID, -         ;Set software volume valid bit.
                     023A   707                  UCB$W_STS(R5)
                     11   023A   708          BRB     DO_FUNCTION             ;Then go do hardware function.
                     023C   709
                     023C   710 UNLOAD:                                       ;UNLOAD
                     023C   711 AVAILABLE:                                    ;AVAILABLE
64 A5     0800 8F    AA   023C   712          BICW2   #UCB$M_VALID, -         ;Clear software volume valid bit.
                     0242   713                  UCB$W_STS(R5)               ;and go complete operation without
                     11   0242   714          BRB     NORMAL                  ;any hardware interaction.
                     0244   715
                     0244   716 READDATA:                                     ;READ DATA
00DD C5     01     88   0244   717          BISB    #CV_M_RD,UCB$B_CV_STS(R5) ;SET READ FLAG
                     0249   718 WRITEDATA:                                    ;WRITE DATA
               007C  30   0249   719          BSBW    FEXL                    ;EXECUTE FUNCTION
                     05'  024C   720          .BYTE   RETRYERR-.-1            ;ERROR OFFSET
                     024D   721 ;
                     024D   722 ;       OPERATON COMPLETION
                     024D   723 ;
                     024D   724
                     024D   725 NORMAL:                                       ;SUCCESSFUL OPERATION COMPLETE
          50   01    3C   024D   726          MOVZWL  #SS$_NORMAL,R0          ;SET NORMAL COMPLETION STATUS
               43    11   0250   727          BRB     FUNCXT                  ;FUNCTION EXIT
                     0252   728
                     0252   729 RETRYERR:                                     ;RETRIABLE ERROR
          0080 C5    97   0252   730          DECB    UCB$B_ERTCNT(R5)        ;ANY RETRIES LEFT?
               03    13   0256   731          BEQL    FATALERR                ;IF EQL - NO
                  FF71   31   0258   732          BRW     FDISPATCH               ;RETRY FUNCTION
                     025B   733
                     025B   734 FATALERR:                                     ;UNRECOVERABLE ERROR
50        0254 8F    3C   025B   735          MOVZWL  #SS$_VOLINV,R0          ;ASSUME VOLUME INVALID STATUS
51        00CC C5    7D   0260   736          MOVQ    UCB$L_CV_CS(R5),R1      ;GET CS IN R1 AND MP IN R2
  2C 52     09     E0   0265   737          BBS     #CV_MP_V_VC,R2,FUNCXT   ;IF SET - VOLUME INVALID
                     0269   738
50        025C 8F    3C   0269   739          MOVZWL  #SS$_WRITLCK,R0         ;ASSUME WRITE LOCK ERROR STATUS
  04 52     0D     E1   026E   740          BBC     #CV_MP_V_WL,R2,10$      ;IF CLR - VOLUME NOT WRITE LOCKED
  1F 52     0A     E0   0272   741          BBS     #CV_MP_V_WGE,R2,FUNCXT  ;IF SET - WRITE GATE ERROR
                     0276   742                                             ;IF WL & WGE SET - WRITE LOCK ERROR
                     0276   743
50        01F4 8F    3C   0276   744 10$:     MOVZWL  #SS$_PARITY,R0          ;ASSUME PARITY ERROR STATUS
```

```
                16 51    0B   E0  027B   745          BBS     #CV_CS_V_CRC,R1,FUNCXT  ;IF SET - CRC ERROR
                                 027F   746                                          ;OR DATAPATH PURGE ERROR
                04 51    0A   E1  027F   747          BBC     #CV_CS_V_OPI,R1,20$     ;HEADER NOT FOUND ERROR?
                0E 51    0C   E0  0283   748          BBS     #CV_CS_V_CVT,R1,FUNCXT  ;IF OPI AND CVT SET - YES
                                 0287   749
            50  008C    8F   3C  0287   750 20$:      MOVZWL  #SS$_DRVERR,R0          ;ASSUME DRIVE ERROR STATUS
              05 51      0E   E0  028C   751          BBS     #CV_CS_V_DE,R1,FUNCXT   ;IF SET - DRIVE ERROR
                                 0290   752
            50  0054    8F   3C  0290   753          MOVZWL  #SS$_CTRLERR,R0         ;ASSUME CONTROLLER ERROR STATUS
                                 0295   754
                                 0295   755 FUNCXT:                                 ;FUNCTION EXIT
                        50   DD  0295   756          PUSHL   R0                      ;SAVE FINAL REQUEST STATUS
             00000000'GF    16  0297   757          JSB     G^IOC$DIAGBUFILL        ;FILL DIAGNOSTIC BUFFER IF PRESENT
            0092 C5     0A   91  029D   758          CMPB    #CDF_WRITECHECK,UCB$B_FEX(R5)  ;DRIVE RELATED FUNCTION?
                        13   1A  02A2   759          BGTRU   10$                     ;IF GTRU - YES
            0092 C5     11   91  02A4   760          CMPB    #CDF_AVAILABLE,UCB$B_FEX(R5)   ;DRIVE RELATED FUNCTION?
                        0C   13  02A9   761          BEQL    10$                     ;IF EQL  - YES
              53    58  A5   D0  02AB   762          MOVL    UCB$L_IRP(R5),R3        ;RETRIEVE ADDRESS OF IRP
                  00C0  C5   A1  02AF   763          ADDW3   UCB$W_BCR(R5),-         ;CALCULATE BYTES TRANSFERRED
            02 AE    32  A3           02B3   764          IRP$W_BCNT(R3),2(SP)    ;...
                                 02B7   765 10$:      RELCHAN                         ;RELEASE CHANNEL IF OWNED
                                 02BD   766
                        51   D4  02BD   767          CLRL    R1                      ;CLEAR SECOND STATUS LONGWORD
                        50 8ED0  02BF   768          POPL    R0                      ;RETRIEVE FINAL REQUEST STATUS
                                 02C2   769          REQCOM                          ;COMPLETE REQUEST
```

```
                     02C8   771
                     02C8   772 ;
                     02C8   773 ; FEXL - RL11 HARDWARE FUNCTION EXECUTION
                     02C8   774 ;
                     02C8   775 ; THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT
                     02C8   776 ; SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN
                     02C8   777 ; SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED
                     02C8   778 ; INTO THE CONSOLE STXCS AND THE FUNCTION IS INITIATED. THE RETURN ADDRESS
                     02C8   779 ; IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE
                     02C8   780 ; INTERRUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
                     02C8   781 ;
                     02C8   782 ; INPUTS:
                     02C8   783 ;
                     02C8   784 ;       R5 = DEVICE UNIT UCB ADDRESS
                     02C8   785 ;
                     02C8   786 ;       00(SP) = RETURN ADDRESS OF CALLER
                     02C8   787 ;       04(SP) = RETURN ADDRESS OF CALLER'S CALLER
                     02C8   788 ;
                     02C8   789 ;       IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS
                     02C8   790 ;       A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.
                     02C8   791 ;
                     02C8   792 ; OUTPUTS:
                     02C8   793 ;
                     02C8   794 ;       THERE ARE FOUR EXITS FROM THIS ROUTINE:
                     02C8   795 ;
                     02C8   796 ;       1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS
                     02C8   797 ;             OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE
                     02C8   798 ;             ERROR ROUTINE.  IN THE CASE OF A POWER FAILURE, THE RETRY
                     02C8   799 ;             COUNT IS RESET AND RETRIES INITIATED.  IN THE CASE OF A
                     02C8   800 ;             TIMEOUT, THE RETRY COUNT IS DECREMENTED AND RETRIES INITIATED
                     02C8   801 ;             IF RETRIES REMAIN.
                     02C8   802 ;
                     02C8   803 ;       2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE
                     02C8   804 ;             ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS EITHER
                     02C8   805 ;             INHIBITED OR EXHAUSTED. IT IS A JUMP TO THE FATAL ERROR EXIT
                     02C8   806 ;             ROUTINE.
                     02C8   807 ;
                     02C8   808 ;       3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER
                     02C8   809 ;             OR DRIVE ERROR OCCURS AND ERROR RETRY IS NEITHER INHIBITED
                     02C8   810 ;             NOR EXHAUSTED. IT CONSISTS OF TAKING THE ERROR BRANCH EXIT
                     02C8   811 ;             SPECIFIED AT THE CALL SITE.  RETRIES ARE ACCOMPLISHED BY
                     02C8   812 ;             RESTARTING THE ENTIRE I/O OPERATION, RATHER THAN AT THE
                     02C8   813 ;             BLOCK IN ERROR.
                     02C8   814 ;
                     02C8   815 ;       4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERRORS OCCUR
                     02C8   816 ;             DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.
                     02C8   817 ;
                     02C8   818 ;       IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.
                     02C8   819 ;
                     02C8   820 ;       IN ALL CASES FINAL DEVICE REGISTERS ARE RETURNED VIA THE UCB.
                     02C8   821 ;
                     02C8   822 ;       UCB$W_BCR(R5) = NEGATIVE BYTES REMAINING TO TRANSFER
                     02C8   823 ;
                     02C8   824 FEXL:                                        ;FUNCTION EXECUTOR
        009C C5 8ED0 02C8   825         POPL    UCB$L_DPC(R5)               ;SAVE DRIVER PC VALUE
50        24 A5   D0 02CD   826         MOVL    UCB$L_CRB(R5),R0            ;GET ADDRESS OF PRIMARY CRB
51        2C A0   D0 02D1   827         MOVL    CRB$L_INTD+VEC$L_IDB(R0),R1 ;GET ADDRESS OF IDB
```

```
                04 A1    55   D1  02D5   828          CMPL    R5,IDB$L_OWNER(R1)      ;DOES THIS PROCESS OWN CHANNEL?
                         05   12  02D9   829          BNEQ    10$                     ;IF NEQ - NO
                   54    61   D0  02DB   830          MOVL    IDB$L_CSR(R1),R4        ;SET ASSIGNED CHANNEL CSR ADDRESS
                         06   11  02DE   831          BRB     20$                     ;
                                  02E0   832 10$:     REQPCHAN                        ;REQUEST CHANNEL (RETURNS R4 = CSR ADR)
                                  02E6   833
                   09    53   91  02E6   834 20$:     CMPB    R3,#CDF_SEARCH          ;TRANSFER FUNCTION?
                         39   1A  02E9   835          BGTRU   XFER                    ;BRANCH IF YES
                                  02EB   836 ;
                                  02EB   837 ; IMMEDIATE FUNCTION EXECUTION
                                  02EB   838 ;
                                  02EB   839 ;       FUNCTIONS INCLUDE:
                                  02EB   840 ;
                                  02EB   841 ;               NO OPERATION,
                                  02EB   842 ;               DRIVE CLEAR, AND
                                  02EB   843 ;               PACK ACKNOWLEDGE
                                  02EB   844 ;
                                  02EB   845 ; INPUTS:
                                  02EB   846 ;       R5      - UCB ADDRESS
                                  02EB   847 ;
                                  02EB   848 ; FUNCTIONAL DESRIPTION:
                                  02EB   849 ;
                                  02EB   850 ; INTERRUPTS ARE LOCKED OUT, THE APPROPRIATE FUNCTION IS INITIATED WITH
                                  02EB   851 ; INTERRUPT ENABLE, AND A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.
                                  02EB   852 ;
                                  02EB   853
                                  02EB   854 IMMED:                                   ;IMMEDIATE FUNCTION EXECUTION
                                  02EB   855          DSBINT                          ;DISABLE INTERRUPTS
                06 64 A5    05    E1  02F1   856          BBC     #UCB$V_POWER,UCB$W_STS(R5),20$ ;BRANCH IF NOT POWERFAIL
                                  02F6   857          ENBINT                          ;POWER FAIL
                         008D    31  02F9   858          BRW     RETREG                  ;PROCESS POWER FAILURE
                                  02FC   859
                00DD C5    02    88  02FC   860 20$:     BISB2   #CV_M_STSONLY,UCB$B_CV_STS(R5) ;REQUEST STATUS ONLY
                00DC C5    01    90  0301   861          MOVB    #ITC_STS1,UCB$B_CV_STATE(R5)    ;SET STATE TO GETSTS1
                     50      44 8F  9A  0306   862          MOVZBL  #<READ_STATUS!STXCS_M_IE>,R0 ;LOAD THE FUNCTION
            0000004C 8F      50    DA  030A   863          MTPR    R0,#PR$_STXCS           ;REQUEST STATUS
                                  0311   864          WFIKPCH RETREG,#59              ;*** for debugging... ;WAITFOR INTERRUPT
                                  031B   865          IOFORK                          ;RETURN FROM ISR-
                                  0321   866                                          ;CREATE FORK PROCESS (&JSB BACK TO ISR)
                         0065    31  0321   867          BRW     RETREG                  ;
```

```
                     0324   869 ;
                     0324   870 ; TRANSFER FUNCTION EXECUTION
                     0324   871 ;
                     0324   872 ;       FUNCTIONS INCLUDE:
                     0324   873 ;
                     0324   874 ;               WRITE DATA
                     0324   875 ;               READ DATA
                     0324   876 ;
                     0324   877 ; INPUTS:
                     0324   878 ;       R4      - DEVICE CSR ADDRESS
                     0324   879 ;       R5      - UCB ADDRESS
                     0324   880 ;
                     0324   881 ; FUNCTIONAL DESCRIPTION:
                     0324   882 ;
                     0324   883 ; THE TRANSFER PARAMETERS ARE LOADED INTO THE CONSOLE REGISTER,
                     0324   884 ; INTERRUPTS ARE LOCKED OUT, THE FUNCTION IS INITIATED, AND
                     0324   885 ; A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.
                     0324   886 ;
                     0324   887 ; UPON RETURN FROM THE INTERRUPT SERVICE ROUTINE, THE TRANSFER WILL
                     0324   888 ; EITHER BE COMPLETE OR AN ERROR WILL HAVE BEEN DETECTED.
                     0324   889 ;
                     0324   890
                     0324   891 XFER:                                   ;TRANSFER FUNCTION EXECUTION
                     0324   892 ;
                     0324   893 ; EXECUTE THE TRANSFER FUNCTION
                     0324   894 ;
                     0324   895         DSBINT
06 64 A5    05   E1  032A   896         BBC     #UCB$V_POWER,UCB$W_STS(R5),20$ ;BRANCH IF NOT POWERFAIL
                     032F   897         ENBINT
          0054   31  0332   898         BRW     RETREG
                     0335   899 ;
                     0335   900 ; SET UP CONTENTS OF STXCS, AND SET MOVE ROUTINE ADDRESS
                     0335   901 ; FOR USE IN INTERRUPT ROUTINE.
                     0335   902 ;
      53    46 8F   9A  0335   903 20$:    MOVZBL  #<READ_BLOCK!STXCS_M_IE>,R3 ;ASSUME READING
   50   00000000'GF   9E  0339   904         MOVAB   G^IOC$MOVTOUSER,R0      ;SET MOVE ROUTINE ADDRESS
    0B 00DD C5    00   E0  0340   905         BBS     #CV_V_RD,UCB$B_CV_STS(R5),40$ ;BRANCH IF READING
      53    45 8F   9A  0346   906         MOVZBL  #<WRITE_BLOCK!STXCS_M_IE>,R3 ;SET FOR WRITING
   50   00000000'GF   9E  034A   907         MOVAB   G^IOC$MOVFRUSER,R0      ;SET MOVE ROUTINE ADDRESS
  00DE C5    0100 8F   B0  0351   908 40$:    MOVW    #256,UCB$W_CV_BBC(R5)   ;SET WORD COUNT
      00E4 C5    50   D0  0358   909         MOVL    R0,UCB$L_CV_MVRTN(R5)   ;SAVE MOVE ROUTINE ADDRESS
                     035D   910 ;
                     035D   911 ; SET LBN INTO R3
                     035D   912 ;
53  10   08    00BC C5   F0  035D   913         INSV    UCB$L_MEDIA(R5),#STXCS_V_ADDRS,#STXCS_S_ADDRS,R3 ;
        50    7E A5   3C  0364   914         MOVZWL  UCB$W_BCNT(R5),R0       ;GET BYTE COUNT OF TRANSFER
   50   00000200 8F   C6  0368   915         DIVL2   #512,R0                 ;COMPUTE # BLOCKS
           50    02   C0  036F   916         ADDL2   #2,R0                   ;THROW IN 2 EXTRA FOR GOOD LUCK
   0000004C 8F    53   DA  0372   917         MTPR    R3,#PR$_STXCS           ;READ/WRITE REQUEST
                     0379   918                                             ;ISR WILL NOT RETURN UNTIL COMPLETE
                     0379   919                                             ;TRANSFER DONE OR ERROR DETECTED.
                     0379   920         WFIKPCH RETREG,R0                   ;WAITFOR INTERRUPT AND KEEP CHANNEL
                     0383   921                                             ;RETURN HERE FROM ISR SAVING REGISTERS
                     0383   922         IOFORK                              ;CREATE FORK PROCESS (RETURN TO ISR)
                     0389   923                                             ;RETURN HERE FROM ISR REI ROUTINE
                     0389   924 ;
                     0389   925 ; GET STATUS AND RESET ERRORS
```

```
                                                C 6
                     0389   926 ;
                     0389   927 RETREG:                                    ;GET STATUS AND RESET ERRORS
                     0389   928 ;
                     0389   929 ; DETERMINE EXIT - SPECIAL CONDITION, FATAL ERROR, RETRIABLE ERROR, OR SUCCESS
                     0389   930 ;
                     0389   931         SETIPL  UCB$B_FIPL(R5)              ;ENSURE AT FORK IPL
 7D 00DD C5   02  E4 038D   932         BBSC    #CV_V_STSERROR,UCB$B_CV_STS(R5),260$ ;BRANCH IF GETSTS ERROR
 73 00DD C5   03  E4 0393   933         BBSC    #CV_V_ABORT,UCB$B_CV_STS(R5),240$ ;BRANCH IF ISR SAID TO ABORT
    02   00DC C5     91 0399   934         CMPB    UCB$B_CV_STATE(R5),#ITC_STS2 ;DID WE GET STATUS?
                 1A  12 039E   935         BNEQ    20$                         ;NO, MUST BE POWERFAIL OR TIMEOUT
    50   00CC C5     7D 03A0   936         MOVQ    UCB$L_CV_CS(R5),R0          ;GET CS AND MP REGISTERS IN R0/R1
1D  51   05   00     ED 03A5   937         CMPZV   #0,#5,R1,-                  ;HEADS, BRUSHES, STATE OK?
                           03AA   938                 #<CV_MP_M_BH!CV_MP_M_HO!CV_SLM>  ;...
                 0E  13 03AA   939         BEQL    20$                         ;IF EQL - YES, ONLINE
 64 A5   0040 8F     AA 03AC   940         BICW2   #UCB$M_TIMOUT,UCB$W_STS(R5)  ;CLEAR DEVICE TIME OUT
    50   01A4 8F     3C 03B2   941         MOVZWL  #SS$_MEDOFL,R0              ;SET MEDIUM OFFLINE STATUS
           FEDB  31     03B7   942         BRW     FUNCRT                      ;RETURN
 64 A5   0060 8F     B3 03BA   943 20$:     BITW    #UCB$M_POWER!-              ;POWER FAIL OR DEVICE TIMEOUT?
                           03C0   944                 UCB$M_TIMOUT,UCB$W_STS(R5)  ;...
              03  13 03C0   945         BEQL    30$                         ;IF EQL NO
            004F  31 03C2   946         BRW     SPECOND                     ;YES - SPECIAL CONDITION
                           03C5   947
   3E 51    09  E0 03C5   948 30$:     BBS     #CV_MP_V_VC,R1,200$         ;IF SET - VOLUME INVALID
   32 50    0F  E1 03C9   949         BBC     #CV_CS_V_CE,R0,100$         ;IF CLEAR RL11 OK
 00000000'GF   16 03CD   950 40$:     JSB     G^ERL$DEVICERR              ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
2E 009A C5    0F  E0 03D3   951         BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),200$  ;IF SET - RETRY INHIBITED
   50   00CC C5     7D 03D9   952         MOVQ    UCB$L_CV_CS(R5),R0          ;GET CS AND MP REGISTERS IN R0/R1
   25 50    0D  E0 03DE   953         BBS     #CV_CS_V_NXM,R0,200$        ;IF SET - NONEXISTENT MEMORY
   0F 50    0E  E1 03E2   954         BBC     #CV_CS_V_DE,R0,80$          ;IF CLR - NO DRIVE ERRORS
   04 51    0D  E1 03E6   955         BBC     #CV_MP_V_WL,R1,60$          ;IF CLR - NOT WRITE LOCKED
   19 51    0A  E0 03EA   956         BBS     #CV_MP_V_WGE,R1,200$        ;IF WL & WGE SET - WL ERROR
   51   C500 8F     B3 03EE   957 60$:     BITW    #<CV_MP_M_WDE!-             ;WRITE DATA ERROR, OR
                           03F3   958                 CV_MP_M_CHE!-               ;CURRENT HEAD ERROR, OR
                           03F3   959                 CV_MP_M_WGE!-               ;WRITE GATE ERROR, OR
                           03F3   960                 CV_MP_M_DSE>,R1             ;DRIVE SELECT ERROR?
            12  12 03F3   961         BNEQ    200$                        ;IF NEQ - YES
                           03F5   962 ;
                           03F5   963 ; RETRIABLE ERROR EXIT
                           03F5   964 ;
7E   009C D5     98 03F5   965 80$:     CVTBL   @UCB$L_DPC(R5),-(SP)        ;GET BRANCH DISPLACEMENT
009C C5     8E   C0 03FA   966         ADDL2   (SP)+,UCB$L_DPC(R5)         ;CALCULATE RETURN ADDRESS - 1
                           03FF   967 ;
                           03FF   968 ; SUCCESSFUL OPERATION EXIT
                           03FF   969 ;
     009C C5     D6 03FF   970 100$:    INCL    UCB$L_DPC(R5)               ;ADJUST TO CORRECT RETURN ADDRESS
     009C D5     17 0403   971         JMP     @UCB$L_DPC(R5)              ;RETURN TO DRIVER
                           0407   972 ;
                           0407   973 ; FATAL ERROR EXIT
                           0407   974 ;
                           0407   975 200$:
            5A  10 0407   976         BSBB    ABORT_RESET_STATUS          ;DO AN ABORT AND RESET STATUS
          FE4F  31 0409   977         BRW     FATALERR                    ;FATAL ERROR EXIT
                           040C   978 ;
                           040C   979 ; ISR DETECTED ERROR.  TELL CONSOLE TO ABORT, AND TRY AGAIN IF WE CAN
                           040C   980 ;
            55  10 040C   981 240$:    BSBB    ABORT_RESET_STATUS          ;ABORT AND RESET STATUS
            E5  11 040E   982         BRB     80$                         ;TRY AGAIN IF RETRIES LEFT
```

```
                                                D 6
                     0410   983 ;
                     0410   984 ; CONSOLE REPORTED ERROR DURING GET STATUS INTERRUPT
                     0410   985 ;
           4A   10   0410   986 260$:   BSBB    RESET_STATUS_ONLY       ;RESET STATUS ONLY
           E1   11   0412   987         BRB     80$                     ;TRY AGAIN IF RETRIES LEFT
                     0414   988 ;
                     0414   989 ; SPECIAL CONDITION EXIT (POWER FAILURE OR DEVICE TIMEOUT)
                     0414   990 ;
                     0414   991 SPECOND:
        2B 64 A5 05  E0   0414   992         BBS     #UCB$V_POWER,UCB$W_STS(R5),PWRFAIL  ;IF SET - POWER FAILURE
                     0419   993                                             ;IF CLR - DEVICE TIMEOUT
     00000000'GF 16   0419   994         JSB     G^ERL$DEVICTMO          ;LOG DEVICE TIMEOUT
  64 A5   0040 8F AA 041F   995         BICW2   #UCB$M_TIMOUT,UCB$W_STS(R5)  ;CLEAR TIMEOUT STATUS
     50   022C 8F 3C 0425   996         MOVZWL  #SS$_TIMEOUT,R0         ;SET DEVICE TIMEOUT STATUS
          0080 C5 97 042A   997         DECB    UCB$B_ERTCNT(R5)        ;ANY ERROR RETRIES REMAINING?
               05 13 042E   998         BEQL    RESETXFR                ;IF EQL - NO
               31 10 0430   999         BSBB    ABORT_RESET_STATUS      ;ABORT AND RESET STATUS
             FD97 31 0432  1000         BRW     FDISPATCH               ;RETRY FUNCTION AGAIN
                     0435  1001
                     0435  1002 RESETXFR:                                 ;RESET TRANSFER BYTE COUNT
               2C 10 0435  1003         BSBB    ABORT_RESET_STATUS      ;ABORT TRANSFER AND RESET STATUS
        53   58 A5 D0 0437  1004         MOVL    UCB$L_IRP(R5),R3        ;GET ADDRESS OF I/O PACKET
  00C0 C5   32 A3 AE 043B  1005         MNEGW   IRP$W_BCNT(R3),UCB$W_BCR(R5)  ;RESET BYTE COUNT
            FE51 31 0441  1006         BRW     FUNCXT                  ;EXIT
                     0444  1007
                     0444  1008 PWRFAIL:                                 ;POWER FAILURE
        64 A5   20 AA 0444  1009         BICW2   #UCB$M_POWER,UCB$W_STS(R5)  ;CLEAR POWER FAILURE BIT
               19 10 0448  1010         BSBB    ABORT_RESET_STATUS      ;ABORT AND RESET STATUS
                  044A  1011         RELCHAN                         ;RELEASE CHANNEL IF OWNED
        53   58 A5 D0 0450  1012         MOVL    UCB$L_IRP(R5),R3        ;GET ADDRESS OF I/O PACKET
             2C A3 7D 0454  1013         MOVQ    IRP$L_SVAPTE(R3),-      ;RESTORE TRANSFER PARAMETERS
             78 A5          0457  1014                 UCB$L_SVAPTE(R5)    ;...
            FD0F 31 0459  1015         BRW     PREPROCESS              ;RETURN TO PREPROCESS UCB FIELDS
                     045C  1016 ;
                     045C  1017 ; ISSUE AN ABORT TO THE CONSOLE.  WHEN THE ABORT COMPLETES, READ
                     045C  1018 ; THE RL11 STATUS REGISTERS, ASSERTING RST.
                     045C  1019 ;
                     045C  1020 ; THIS ROUTINE DESTROYS R0-R3
                     045C  1021 ;
                     045C  1022         .ENABLE LOCAL_BLOCK
                     045C  1023
                     045C  1024 RESET_STATUS_ONLY:
          00F8 C5 8ED0 045C  1025         POPL    UCB$L_CV_ABPC(R5)       ;POP RETURN ADDRESS FROM STACK
               2B 11 0461  1026         BRB     30$                     ;GO EXECUTE
                     0463  1027
                     0463  1028 ABORT_RESET_STATUS:
          00F8 C5 8ED0 0463  1029         POPL    UCB$L_CV_ABPC(R5)       ;POP RETURN ADDRESS FROM STACK
     00DC C5   03 90 0468  1030         MOVB    #ITC_ABORT,UCB$B_CV_STATE(R5) ;SET DISPATCH
                     046D  1031         DSBINT                          ;DISABLE INTERRUPTS
        50   43 8F 9A 0473  1032         MOVZBL  #<ABORT_TRANSFER!STXCS_M_IE>,R0 ;SETUP FUNCTION
 0000004C 8F   50 DA 0477  1033         MTPR    R0,#PR$_STXCS           ;TELL THE CONSOLE TO ABORT
                     047E  1034         WFIKPCH 20$,#6
                     0488  1035 20$:     IOFORK
  00D4 C5   00CC C5 7D 048E  1036 30$:     MOVQ    UCB$L_CV_CS(R5),UCB$Q_CV_CSMP(R5) ;SAVE CS/MP REGISTERS
                     0495  1037         TIMEDWAIT TIME=#600*1000,-      ;WAIT FOR CONSOLE TO BE READY
                     0495  1038                 INS1=<MFPR      #PR$_STXCS,R2>,- ;READ STATUS REGISTER
                     0495  1039                 INS2=<BBS       #STXCS_V_RDY,R2,40$>,- ;BRANCH IF READY
```

```
                                0495  1040                    DONELBL=40$                             ;TO SAME PLACE AS DONE
                                04C2  1041 ;
                                04C2  1042 ; ** WHAT DO WE DO IF THE CONSOLE DOES NOT GO READY IN TIME?
                                04C2  1043 ;
                00DC C5   04    90  04C2  1044          MOVB    #ITC_RESET1,UCB$B_CV_STATE(R5) ;SET DISPATCH
                                04C7  1045          DSBINT                                  ;DISABLE INTERRUPTS
    0000004C 8F    00000042 8F  DA  04CD  1046          MTPR    #<STATUS_RESET!STXCS_M_IE>,#PR$_STXCS ;REQUEST STATUS WITH RST ASSER
                                04D8  1047          WFIKPCH 60$,#6                          ;WAITFOR INTERRUPT
                                04E2  1048 60$:     IOFORK
            00CC C5    00D4 C5  7D  04E8  1049          MOVQ    UCB$Q_CV_CSMP(R5),UCB$L_CV_CS(R5) ;RESTORE CS/MP REGISTERS
                       00F8 D5  17  04EF  1050          JMP     @UCB$L_CV_ABPC(R5)              ;RETURN TO CALLER
                                04F3  1051
                                04F3  1052          .DISABLE LOCAL_BLOCK
```

```
                                04F3  1054          .SBTTL  INTERRUPT SERVICE ROUTINE
                                04F3  1055 ;++
                                04F3  1056 ; CV$INT - VAX 8600 CONSOLE RL02 INTERRUPT SERVICE ROUTINE
                                04F3  1057 ;
                                04F3  1058 ; FUNCTIONAL DESCRIPTION:
                                04F3  1059 ;
                                04F3  1060 ;       THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT
                                04F3  1061 ;       OCCURS ON THE VAX 8600 CONSOLE STXCS REGISTER. IF THE INTERRUPT
                                04F3  1062 ;       IS NOT EXPECTED, THE UNSOLICITED INTERRUPT ROUTINE DISMISSES
                                04F3  1063 ;       THE INTERRUPT. IF THE INTERRUPT IS EXPECTED, DEVICE REGISTERS
                                04F3  1064 ;       ARE SAVED AND THE DRIVER IS CALLED AT ITS INTERRUPT RETURN ADDRESS.
                                04F3  1065 ;       THE DRIVER FORKS, CAUSING A RETURN TO THIS ROUTINE,
                                04F3  1066 ;       WHICH RESTORES GENERAL REGISTERS AND DISMISSES THE INTERRUPT.
                                04F3  1067 ;
                                04F3  1068 ; INPUTS:
                                04F3  1069 ;
                                04F3  1070 ;       00(SP)  - POINTER TO ADDRESS OF THE IDB
                                04F3  1071 ;       04(SP)  - SAVED R0
                                04F3  1072 ;       08(SP)  - SAVED R1
                                04F3  1073 ;       12(SP)  - SAVED R2
                                04F3  1074 ;       16(SP)  - SAVED R3
                                04F3  1075 ;       20(SP)  - SAVED R4
                                04F3  1076 ;       24(SP)  - SAVED R5
                                04F3  1077 ;       28(SP)  - PC AT THE TIME OF THE INTERRUPT
                                04F3  1078 ;       32(SP)  - PSL AT THE TIME OF THE INTERRUPT
                                04F3  1079 ;
                                04F3  1080 ; OUTPUTS:
                                04F3  1081 ;
                                04F3  1082 ;       DEVICE REGISTERS ARE SAVED, IPL IS LOWERED TO FORK LEVEL, THE
                                04F3  1083 ;       INTERRUPT IS DISMISSED, ALL REGISTERS EXCEPT R0-R5 ARE PRESERVED.
                                04F3  1084 ;
                                04F3  1085 ;--
                                04F3  1086
                                04F3  1087 CV_INT::                                     ;INTERRUPT SERVICE ROUTINE
          53    9E    D0        04F3  1088          MOVL    @(SP)+,R3                 ;REMOVE ADDRESS OF IDB FROM STACK
          54    63    7D        04F6  1089          MOVQ    (R3),R4                   ;GET ADDRESS OF CSR AND UCB
                55    D5        04F9  1090          TSTL    R5                        ;IS R5 A ZERO
                27    13        04FB  1091          BEQL    CV_UNSOLNT                ;IF EQL NO OWNER
53    0000004C    8F    DB        04FD  1092          MFPR    #PR$_STXCS,R3             ;**TEMP** READ CONSOLE STATUS
        1C 53     07    E1        0504  1093          BBC     #STXCS_V_RDY,R3,CV_UNSOLNT ;**TEMP** BRANCH IF NOT READY
                  01    E5        0508  1094          BBCC    #UCB$V_INT,-              ;IF CLR - INTERRUPT NOT EXPECTED
             17 64    A5          050A  1095          UCB$W_STS(R5),CV_UNSOLNT  ;...
                                050D  1096 CV_INT_DISP:
      53    00DC    C5    9A        050D  1097          MOVZBL  UCB$B_CV_STATE(R5),R3     ;GET INTERRUPT STATE
                40    13        0512  1098          BEQL    CV_INT_XFR                ;BRANCH IF TRANSFER INTERRUPT
                                0514  1099          CASE    R3,<-                     ; AND DISPATCH
                                0514  1100                  CV_INT_XFR,-              ;TRANSFER INTERRUPT
                                0514  1101                  CV_INT_STS1,-             ;FIRST PART OF STATUS
                                0514  1102                  CV_INT_STS2,-             ;SECOND PART OF STATUS
                                0514  1103                  CV_INT_ABORT,-            ;ABORT REQUEST
                                0514  1104                  CV_INT_RSTS1,-            ;GET STATUS WITH RST ASSERTED
                                0514  1105                  CV_INT_RSTS2>,-
                                0514  1106                  TYPE=B
                                0524  1107
                                0524  1108 CV_UNSOLNT:                                ;UNSOLICITED INTERRUPT
                   3F    BA        0524  1109          POPR    #^M<R0,R1,R2,R3,R4,R5>    ;RESTORE R0-R5
                   02    02        0526  1110          REI                               ;RETURN FROM INTERRUPT
```

```
                                0527  1111 ;
                                0527  1112 ; GET STATUS WITH RESET INTERRUPT
                                0527  1113 ;
                                0527  1114 CV_INT_RSTS1:
      00CC C5   0000004D 8F   DB  0527  1115          MFPR    #PR$_STXDB,UCB$L_CV_CS(R5) ;READ CONTROL/STATUS REGISTER
              00DC C5    05   90  0530  1116          MOVB    #ITC_RESET2,UCB$B_CV_STATE(R5) ;SET NEXT STATE
  0000004C 8F   00000042 8F   DA  0535  1117          MTPR    #<STATUS_RESET!STXCS_M_IE>,#PR$_STXCS
               64 A5    02   88  0540  1118          BISB2   #UCB$M_INT,UCB$W_STS(R5) ;FLAG INTERRUPT EXPTECTED
                        DE   11  0544  1119          BRB     CV_UNSOLNT
                                0546  1120 CV_INT_RSTS2:
      00D0 C5   0000004D 8F   DB  0546  1121          MFPR    #PR$_STXDB,UCB$L_CV_MP(R5) ;SAVE MULTIPURPOSE REGISTER
                                054F  1122 CV_INT_ABORT:
                     0C B5   16  054F  1123          JSB     @UCB$L_FPC(R5)          ;CALL DRIVER AT INTERRUPT RETURN ADDR
                        D0   11  0552  1124          BRB     CV_UNSOLNT
                                0554  1125 ;
                                0554  1126 ; TRANSFER INTERRUPT
                                0554  1127 ;
                                0554  1128 CV_INT_XFR:
         53   0000004C 8F   DB  0554  1129          MFPR    #PR$_STXCS,R3           ;GET STATUS REGISTER
        53   53    08   18   EF  055B  1130          EXTZV   #STXCS_V_STS,#STXCS_S_STS,R3,R3 ;GET CONSOLE RL02 STATUS
                  53   02   D1  0560  1131          CMPL    #TRANS_CONTINUE,R3      ;CONTINUE TRANSACTION?
                       03   13  0563  1132          BEQL    10$                     ;YES, CONTINUE
                     00CE   31  0565  1133          BRW     400$                    ;ELSE BRANCH TO ABORT
          64 00DD C5   00   E1  0568  1134 10$:     BBC     #CV_V_RD,UCB$B_CV_STS(R5),200$ ;BRANCH IF WRITING
                                056E  1135 ;
                                056E  1136 ; OPERATION IS A READ FROM DISK
                                056E  1137 ;
          7E   0000004D 8F   DB  056E  1138          MFPR    #PR$_STXDB,-(SP)        ;READ DATA ONTO STACK
                51   46 8F   9A  0575  1139          MOVZBL  #<READ_BLOCK!STXCS_M_IE>,R1 ;SET NEXT READ
    51   10   08   00BC C5   F0  0579  1140          INSV    UCB$L_MEDIA(R5),#STXCS_V_ADDRS,#STXCS_S_ADDRS,R1 ;SET LBN
         0000004C 8F   51   DA  0580  1141          MTPR    R1,#PR$_STXCS           ;CONTINUE READING
                  00C0 C5   B5  0587  1142          TSTW    UCB$W_BCR(R5)           ;HAVE WE COMPLETED THE REQUEST?
                       22   13  058B  1143          BEQL    20$                     ;IF EQL YES, DON'T WRITE TO BUFFER
                  51   5E   D0  058D  1144          MOVL    SP,R1                   ;GET ADDRESS OF DATA
                  52   02   D0  0590  1145          MOVL    #2,R2                   ;WRITE 2 BYTES INTO USER BUFFER
              50  00E8 C5   D0  0593  1146          MOVL    UCB$L_CV_BUFWIN(R5),R0  ;GET BUFFER ADDRESS
                  00E4 D5   16  0598  1147          JSB     @UCB$L_CV_MVRTN(R5)     ;WRITE INTO USER BUFFER
              00E8 C5   50   D0  059C  1148          MOVL    R0,UCB$L_CV_BUFWIN(R5)  ;SAVE WINDOW INTO USER BUFFER
    00E4 C5   00000000'GF   9E  05A1  1149          MOVAB   G^IOC$MOVTOUSER2,UCB$L_CV_MVRTN(R5) ;SET MOVE ROUTINE ADDRESS
              00C0 C5   02   A0  05AA  1150          ADDW2   #2,UCB$W_BCR(R5)        ;COUNT TWO MORE BYTES TRANSFERRED
                  5E   04   C0  05AF  1151 20$:     ADDL2   #4,SP                   ;CLEAR DATA FROM STACK
                  00DE C5   B7  05B2  1152          DECW    UCB$W_CV_BBC(R5)        ;COUNT ANOTHER WORD TRANSFERRED
                       12   19  05B6  1153          BLSS    120$                    ;PROTOCOL ERROR
         54   0000004C 8F   DB  05B8  1154          MFPR    #PR$_STXCS,R4           ;READ STXCS
               91 54    07   E0  05BF  1155          BBS     #STXCS_V_RDY,R4,CV_INT_XFR ;BRANCH IF DONE AGAIN
               64 A5    02   88  05C3  1156          BISB2   #UCB$M_INT,UCB$W_STS(R5);FLAG INTERRUPT EXPECTED
                     FF5A   31  05C7  1157 100$:     BRW     CV_UNSOLNT              ;EXIT THIS INTERRUPT
              00DD C5   08   88  05CA  1158 120$:     BISB2   #CV_M_ABORT,UCB$B_CV_STS(R5) ;FLAG TO ABORT AND RETRY
                     FF7D   31  05CF  1159          BRW     CV_INT_ABORT            ;CALL DRIVER TO DO IT
                                05D2  1160 ;
                                05D2  1161 ; WRITING TO DISK
                                05D2  1162 ;
                  00C0 C5   B5  05D2  1163 200$:    TSTW    UCB$W_BCR(R5)           ;REQUEST COMPLETE?
                       24   13  05D6  1164          BEQL    220$                    ;IF EQL YES DON'T BOTHER FETCHING
              51  00E0 C5   9E  05D8  1165          MOVAB   UCB$L_CV_IBUF(R5),R1    ;GET ADDRESS OF INTERNAL BUFFER
                  52   02   D0  05DD  1166          MOVL    #2,R2                   ;SET NUMBER OF BYTES
              50  00E8 C5   D0  05E0  1167          MOVL    UCB$L_CV_BUFWIN(R5),R0  ;GET BUFFER WINDOW
```

```
                                                                H 6
                00E4 D5   16  05E5  1168          JSB     @UCB$L_CV_MVRTN(R5)     ;GET 2 BYTES FROM USERS BUFFER
           00E8 C5   50   D0  05E9  1169          MOVL    R0,UCB$L_CV_BUFWIN(R5)  ;SAVE WINDOW
      00E4 C5   00000000'GF 9E  05EE  1170          MOVAB   G^IOC$MOVFRUSER2,UCB$L_CV_MVRTN(R5) ;SET MOVE ROUTINE
           00C0 C5   02   A0  05F7  1171          ADDW2   #2,UCB$W_BCR(R5)        ;COUNT TWO MORE BYTES
 0000004D 8F   00E0 C5   DA  05FC  1172 220$:    MTPR    UCB$L_CV_IBUF(R5),#PR$_STXDB ;WRITE WORD TO CONSOLE
           00C0 C5   B5  0605  1173          TSTW    UCB$W_BCR(R5)           ;REQUEST COMPLETE?
                04   12  0609  1174          BNEQ    240$                    ;IF NEQ NO
           00E0 C5   D4  060B  1175          CLRL    UCB$L_CV_IBUF(R5)       ;YES, CLEAR BUFFER SO WE WRITE 0'S
           00DE C5   B7  060F  1176 240$:    DECW    UCB$W_CV_BBC(R5)        ;COUNT ANOTHER WORD TRANSFERRED
                19   19  0613  1177 250$:    BLSS    280$                    ;PROTOCOL ERROR
           51   45 8F   9A  0615  1178          MOVZBL  #<WRITE_BLOCK!STXCS_M_IE>,R1 ;REQUEST TO SEND AGAIN
51   10   08   00BC C5   F0  0619  1179          INSV    UCB$L_MEDIA(R5),#STXCS_V_ADDRS,#STXCS_S_ADDRS,R1 ;SET LBN
      0000004C 8F   51   DA  0620  1180          MTPR    R1,#PR$_STXCS           ;SEND COMMAND TO CONSOLE
           64 A5   02   88  0627  1181 260$:    BISB2   #UCB$M_INT,UCB$W_STS(R5);FLAG INTERRUPT EXPECTED
              FEF6   31   062B  1182          BRW     CV_UNSOLNT              ;DISMISS INTERRUPT
           00DD C5   08   88  062E  1183 280$:    BISB2   #CV_M_ABORT,UCB$B_CV_STS(R5) ;FLAG TO ABORT AND RETRY
              FF19   31   0633  1184          BRW     CV_INT_ABORT            ;CALL DRIVER TO DO IT
                         0636  1185 ;
                         0636  1186 ; TRANSACTION COMPLETE, OR ERROR DETECTED.  REQUEST STATUS
                         0636  1187 ;
           53   80 8F   91  0636  1188 400$:    CMPB    #HANDSHAKE_ERROR,R3     ;WAS THERE A HANDSHAKE ERROR?
                1D   13  063A  1189          BEQL    440$                    ;BRANCH IF YES
           00DE C5   B5  063C  1190          TSTW    UCB$W_CV_BBC(R5)        ;ALL WORDS TRANSFERRED?
                18   12  0640  1191          BNEQ    460$                    ;IF NEQ NO
           00DC C5   01  90  0642  1192 420$:    MOVB    #ITC_STS1,UCB$B_CV_STATE(R5) ;SET NEXT STATE
 0000004C 8F   00000044 8F   DA  0647  1193          MTPR    #<READ_STATUS!STXCS_M_IE>,#PR$_STXCS ;REQUEST STATUS
           64 A5   02   88  0652  1194          BISB2   #UCB$M_INT,UCB$W_STS(R5);FLAG INTERRUPT EXPECTED
              FECB   31   0656  1195          BRW     CV_UNSOLNT              ;DISMISS INTERRUPT
                         0659  1196 ;
                         0659  1197 ; HANDSHAKE ERROR.  TELL DRIVER TO ABORT AND RETRY
                         0659  1198 ;
                      01 0659  1199 440$:    NOP                             ;**DEBUG
                         065A  1200 ;
                         065A  1201 ; NOT ALL WORDS TRANSFERRED.  TELL DRIVER TO ABORT AND RETRY
                         065A  1202 ;
           00DD C5   08   88  065A  1203 460$:    BISB2   #CV_M_ABORT,UCB$B_CV_STS(R5) ;FLAG TO ABORT AND RETRY
              FEED   31   065F  1204          BRW     CV_INT_ABORT            ;CALL DRIVER TO DO IT
                         0662  1205 ;
                         0662  1206 ; ERROR ON GET STATUS OPERATION
                         0662  1207 ;
                         0662  1208 CV_STSERROR:
           00DD C5   04   88  0662  1209          BISB2   #CV_M_STSERROR,UCB$B_CV_STS(R5) ;FLAG GET STATUS ERROR
              FEE5   31   0667  1210          BRW     CV_INT_ABORT            ;CALL DRIVER TO PROCESS ERROR
                         066A  1211 ;
                         066A  1212 ; GET STATUS PART 1 INTERRUPT
                         066A  1213 ;
                         066A  1214 CV_INT_STS1:
      53   0000004C 8F   DB  066A  1215          MFPR    #PR$_STXCS,R3           ;READ STXCS REGISTER
        ED 53   1F   E0  0671  1216          BBS     #31,R3,CV_STSERROR      ;BRANCH IF ERROR GETTING STATUS
 00CC C5   0000004D 3F   DB  0675  1217          MFPR    #PR$_STXDB,UCB$L_CV_CS(R5) ;GET THE CONTROL/STATUS REGISTER
      00DC C5   02   90  067E  1218          MOVB    #ITC_STS2,UCB$B_CV_STATE(R5)    ;SET NEXT STATE
 0000004C 8F   00000044 8F   DA  0683  1219          MTPR    #<READ_STATUS!STXCS_M_IE>,#PR$_STXCS ;REQUEST IT
           64 A5   02   88  068E  1220          BISB2   #UCB$M_INT,UCB$W_STS(R5);FLAG INTERRUPT EXPECTED
              FE8F   31   0692  1221          BRW     CV_UNSOLNT              ;DISMISS INTERRUPT
                         0695  1222 ;
                         0695  1223 ; GET STATUS PART 2 INTERRUPT
                         0695  1224 ;
```

```
                              0695  CV_INT_STS2:
        53   0000004C 8F  DB  0695          MFPR    #PR$_STXCS,R3                   ;READ STXCS REGISTER
                C2 53  1F  E0  069C          BBS     #31,R3,CV_STSERROR              ;BRANCH IF ERROR GETTING STATUS
  00D0 C5   0000004D 8F  DB  06A0          MFPR    #PR$_STXDB,UCB$L_CV_MP(R5) ;GET MULTIPURPOSE REGISTER
     24 00DD C5        01  E0  06A9          BBS     #CV_V_STSONLY,UCB$B_CV_STS(R5),20$ ;BRANCH IF STATUS ONLY
                              06AF  ;
                              06AF  ; TRANSFER OF A BLOCK IS COMPLETE.  SEE IF ERRORS, AND PROCESS IF SO.
                              06AF  ; IF NO ERRORS, THEN SEE IF DONE WITH COMPLETE TRANSFER
                              06AF  ;
        50    00CC C5     7D  06AF          MOVQ    UCB$L_CV_CS(R5),R0              ;GET CS AND MP REGISTERS
  1D    51    05    00    ED  06B4          CMPZV   #0,#5,R1,-                      ;HEADS AND BRUSHES OK?
                              06B9                  #<CV_MP_M_BH!CV_MP_M_HO!CV_SLM>
              18    12        06B9          BNEQ    20$                             ;IF NEQ NO
        50    E000 8F     B3  06BB          BITW    #<CV_CS_M_CE!CV_CS_M_DE!CV_CS_M_NXM>,R0
              11    12        06C0          BNEQ    20$                             ;IF NEQ ERROR OF SOME SORT
        51    C700 8F     B3  06C2          BITW    #<CV_MP_M_WDE!-
                              06C7                  CV_MP_M_CRE!-
                              06C7                  CV_MP_M_WGE!-
                              06C7                  CV_MP_M_DSE!-
                              06C7                  CV_MP_M_VC>,R1                  ;ANY ERRORS?
              0A    12        06C7          BNEQ    20$                             ;IF NEQ YES
         00BC C5    D6        06C9          INCL    UCB$L_MEDIA(R5)                 ;NEXT LBN
         00C0 C5    B5        06CD          TSTW    UCB$W_BCR(R5)                   ;ARE WE DONE YET?
              06    12        06D1          BNEQ    40$                             ;BRANCH IF NOT DONE YET
           0C B5    16        06D3  20$:    JSB     @UCB$L_FPC(R5)                  ;CALL DRIVER AT INTERRUPT RETURN ADDR
            FE4B    31        06D6  30$:    BRW     CV_UNSOLNT                      ;DISMISS
                              06D9  ;
                              06D9  ; MORE DATA TO TRANSFER STILL
                              06D9  ;
   00DE C5    0100 8F     B0  06D9  40$:    MOVW    #256,UCB$W_CV_BBC(R5)           ;RESET BYTE COUNT FOR BLOCK
         00DC C5    94        06E0          CLRB    UCB$B_CV_STATE(R5)              ;RESET STATE TO READ MODE
        53    46 8F       9A  06E4          MOVZBL  #<READ_BLOCK!STXCS_M_IE>,R3 ;ASSUME READING
    04 00DD C5    00      E0  06E8          BBS     #CV_V_RD,UCB$B_CV_STS(R5),60$ ;BRANCH IF READING
        53    45 8F       9A  06EE          MOVZBL  #<WRITE_BLOCK!STXCS_M_IE>,R3 ;SET FOR WRITING
 53   10   08   00BC C5   F0  06F2  60$:    INSV    UCB$L_MEDIA(R5),#STXCS_V_ADDRS,#STXCS_S_ADDRS,R3 ;SET LBN
   0000004C 8F    53      DA  06F9          MTPR    R3,#PR$_STXCS                   ;SEND COMMAND TO CONSOLE
       64 A5    02        88  0700          BISB2   #UCB$M_INT,UCB$W_STS(R5);FLAG INTERRUPT EXPECTED
            FE1D    31        0704          BRW     CV_UNSOLNT                      ;EXIT INTERRUPT
                              0707
                              0707  CVPATCH::
                   00000787   0707          .BLKL   32
```

```
                        0787  1267          .SBTTL  REGISTER DUMP ROUTINE
                        0787  1268 ;++
                        0787  1269 ;
                        0787  1270 ; CV_REGDUMP - REGISTER DUMP ROUTINE
                        0787  1271 ;
                        0787  1272 ; FUNCTIONAL DESCRIPTION:
                        0787  1273 ;
                        0787  1274 ; THIS ROUTINE IS CALLED TO SAVE THE DEVICE REGISTERS AND UBA RESOURCE
                        0787  1275 ; REGISTERS IN A SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR
                        0787  1276 ; LOGGING ROUTINE AND FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.
                        0787  1277 ;
                        0787  1278 ; INPUTS:
                        0787  1279 ;
                        0787  1280 ;       R0      - ADDRESS OF REGISTER SAVE BUFFER
                        0787  1281 ;       R4      - ADDRESS OF DEVICE CONTROL STATUS REGISTER (CSR)
                        0787  1282 ;       R5      - ADDRESS OF UNIT CONTROL BLOCK (UCB)
                        0787  1283 ;
                        0787  1284 ; OUTPUTS:
                        0787  1285 ;
                        0787  1286 ;       THE DEVICE AND UBA REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.
                        0787  1287 ;       R0 CONTAINS THE ADDRESS OF THE NEXT EMPTY LONGWORD IN THE BUFFER.
                        0787  1288 ;       ALL REGISTERS EXCEPT R1 AND R2 ARE PRESERVED.
                        0787  1289 ;
                        0787  1290 ;--
                        0787  1291
                        0787  1292 CV_REGDUMP:                                  ;REGISTER DUMP ROUTINE
     80   09   D0      0787  1293          MOVL    #<CV_NUM_REGS+5>,(R0)+      ;INSERT NUMBER OF REGISTERS
80   00CC C5   3C      078A  1294          MOVZWL  UCB$L_CV_CS(R5),(R0)+       ;COPY CONTROL/STATUS REGISTER
          80   7C      078F  1295          CLRQ    (R0)+                       ;NO BA/DA REGISTERS
80   00D0 C5   3C      0791  1296          MOVZWL  UCB$L_CV_MP(R5),(R0)+       ;COPY MULTIPURPOSE REGISTER
          80   7C      0796  1297          CLRQ    (R0)+                       ;NO DATAPATH NUMBER/DATAPATH REGISTER
          80   7C      0798  1298          CLRQ    (R0)+                       ;NO FINAL MAP REG/PREVIOUS MAP REG
80        80   D4      079A  1299          CLRL    (R0)+                       ;NO DATAPATH PURGE ERROR REGISTER
          05   05      079C  1300          RSB                                 ;RETURN
                        079D  1301
                        079D  1302 CV_END:                                      ;ADDRESS OF LAST LOCATION IN DRIVER
                        079D  1303          .END
```

```
$$$                             = 00000020 R      02
$$GENF_CODE                     = 00000012
$$OP                            = 00000002
ABORT_RESET_STATUS                00000463 R      03
ABORT_TRANSFER                  = 00000003
ACP$ACCESS                        ******** X      03
ACP$DEACCESS                      ******** X      03
ACP$MODIFY                        ******** X      03
ACP$MOUNT                         ******** X      03
ACP$READBLK                       ******** X      03
ACP$WRITEBLK                      ******** X      03
ATS_UBA                         = 00000001
AVAILABLE                         0000023C R      03
CDF_AVAILABLE                   = 00000011
CDF_DRVCLR                      = 00000004
CDF_NOP                         = 00000010
CDF_OFFSET                      = 00000006
CDF_PACKACK                     = 00000008
CDF_READDATA                    = 0000000C
CDF_READHEAD                    = 0000000E
CDF_RECAL                       = 00000003
CDF_RELEASE                     = 00000005
CDF_RETCENTER                   = 00000007
CDF_SEARCH                      = 00000009
CDF_SEEK                        = 00000002
CDF_UNLOAD                      = 00000001
CDF_WRITECHECK                  = 0000000A
CDF_WRITEDATA                   = 0000000B
CDF_WRITEHEAD                   = 0000000D
CRB$L_INTD                      = 00000024
CV$DDT                            00000000 RG     03
CVC_GETSTS                        0000010A R      03
CVPATCH                           00000707 RG     03
CV_ALIGN                          0000015B R      03
CV_CS                             00000000
CV_CS_M_CE                      = 00008000
CV_CS_M_DE                      = 00004000
CV_CS_M_NXM                     = 00002000
CV_CS_V_CE                      = 0000000F
CV_CS_V_CRC                     = 0000000B
CV_CS_V_CVT                     = 0000000C
CV_CS_V_DE                      = 0000000E
CV_CS_V_DRDY                    = 00000000
CV_CS_V_NXM                     = 0000000D
CV_CS_V_OPI                     = 0000000A
CV_END                            0000079D R      03
CV_FUNCTABLE                      00000038 R      03
CV_INT                            000004F3 RG     03
CV_INT_ABORT                      0000054F R      03
CV_INT_DISP                       00000500 R      03
CV_INT_RSTS1                      00000527 R      03
CV_INT_RSTS2                      00000546 R      03
CV_INT_STS1                       0000066A R      03
CV_INT_STS2                       00000695 R      03
CV_INT_XFR                        00000554 R      03
CV_MP                             00000002
CV_MP_M_BH                      = 00000008
CV_MP_M_CHE                     = 00004000
CV_MP_M_DSE                     = 00000100
CV_MP_M_HO                      = 00000010
CV_MP_M_VC                      = 00000200
CV_MP_M_WDE                     = 00008000
CV_MP_M_WGE                     = 00000400
CV_MP_V_VC                      = 00000009
CV_MP_V_WGE                     = 0000000A
CV_MP_V_WL                      = 0000000D
CV_M_ABORT                      = 00000008
CV_M_RD                         = 00000001
CV_M_STSERROR                   = 00000004
CV_M_STSONLY                    = 00000002
CV_NOM_REGS                     = 00000004
CV_REGDUMP                        00000787 R      03
CV_RLOX_INIT                      000000D9 R      03
CV_RL11_INIT                      000000D8 R      03
CV_SLM                          = 00000005
CV_STARTIO                        0000016B R      03
CV_STSERROR                       00000662 R      03
CV_UNSOLNT                        00000524 R      03
CV_V_ABORT                      = 00000003
CV_V_RD                         = 00000000
CV_V_STSERROR                   = 00000002
CV_V_STSONLY                    = 00000001
DC$_DISK                        = 00000001
DDB$K_CART                      = 00000002
DDB$L_ACPD                      = 00000010
DDB$L_DDT                       = 0000000C
DEV$M_AVL                       = 00040000
DEV$M_DIR                       = 00000008
DEV$M_ELG                       = 00400000
DEV$M_FOD                       = 00004000
DEV$M_IDV                       = 04000000
DEV$M_ODV                       = 08000000
DEV$M_RND                       = 10000000
DEV$M_SHR                       = 00010000
DO_FUNCTION                       0000022A R      03
DPT$C_LENGTH                    = 00000038
DPT$C_VERSION                   = 00000004
DPT$INITAB                        00000038 R      02
DPT$M_SVP                       = 00000002
DPT$REINITAB                      00000067 R      02
DPT$TAB                           00000000 R      02
DRVCLR                            0000022A R      03
DT$_RL02                        = 0000000A
DYN$C_CRB                       = 00000005
DYN$C_DDB                       = 00000006
DYN$C_DPT                       = 0000001E
DYN$C_UCB                       = 00000010
EMB$L_DV_REGSAV                 = 0000004E
ERL$DEVICERR                      ******** X      03
ERL$DEVICTMO                      ******** X      03
EXE$ABORTIO                       ******** X      03
EXE$GL_TENUSEC                    ******** X      03
EXE$GL_UBDELAY                    ******** X      03
EXE$IOFORK                        ******** X      03
```

```
EXE$LCLDSKVALID          ********    X   03
EXE$ONEPARM              ********    X   03
EXE$SENSEMODE            ********    X   03
EXE$SETCHAR              ********    X   03
EXE$ZEROPARM             ********    X   03
FATALERR                 0000025B R      03
FDISPATCH                000001CC R      03
FEXL                     000002C8 R      03
FUNCTAB_LEN            = 000000A0
FUNCXT                   00000295 R      03
HANDSHAKE_ERROR        = 00000080
HW_ERROR               = 00000081
IDB$L_CSR              = 00000000
IDB$L_OWNER            = 00000004
IMMED                    000002EB R      03
IO$V_INHRETRY          = 0000000F
IO$_ACCESS             = 00000032
IO$_ACPCONTROL         = 00000038
IO$_AVAILABLE          = 00000011
IO$_CREATE             = 00000033
IO$_DEACCESS           = 00000034
IO$_DELETE             = 00000035
IO$_DRVCLR             = 00000004
IO$_MODIFY             = 00000036
IO$_MOUNT              = 00000039
IO$_NOP                = 00000000
IO$_PACKACK            = 00000008
IO$_READLBLK           = 00000021
IO$_READPBLK           = 0000000C
IO$_READVBLK           = 00000031
IO$_SEEK               = 00000002
IO$_SENSECHAR          = 0000001B
IO$_SENSEMODE          = 00000027
IO$_SETCHAR            = 0000001A
IO$_SETMODE            = 00000023
IO$_UNLOAD             = 00000001
IO$_VIRTUAL            = 0000003F
IO$_WRITELBLK          = 00000020
IO$_WRITEPBLK          = 0000000B
IO$_WRITEVBLK          = 00000030
IOC$DIAGBUFILL           ********    X   03
IOC$MNTVER               ********    X   03
IOC$MOVFRUSER            ********    X   03
IOC$MOVFRUSER2           ********    X   03
IOC$MOVTOUSER            ********    X   03
IOC$MOVTOUSER2           ********    X   03
IOC$RELCHAN              ********    X   03
IOC$REQCOM               ********    X   03
IOC$REQPCHANL            ********    X   03
IOC$RETURN               ********    X   03
IOC$WFIKPCH              ********    X   03
IRP$L_MEDIA            = 00000038
IRP$L_SVAPTE           = 0000002C
IRP$S_FCODE            = 00000006
IRP$V_DIAGBUF          = 00000007
IRP$V_FCODE            = 00000000
IRP$V_PHYSIO           = 00000008
IRP$W_BCNT             = 00000032
IRP$W_FUNC             = 00000020
IRP$W_STS              = 0000002A
ITC_ABORT              = 00000003
ITC_DATA               = 00000000
ITC_RESET1             = 00000004
ITC_RESET2             = 00000005
ITC_STS1               = 00000001
ITC_STS2               = 00000002
MASKH                  = 00000008
MASKL                  = 04000000
NOP                      0000022A R      03
NORMAL                   0000024D R      03
NO_OP                  = 00000000
PACKACK                  00000234 R      03
PR$_IPL                = 00000012
PR$_STXCS              = 0000004C
PR$_STXDB              = 0000004D
PREPROCESS               0000016B R      03
PWRFAIL                  00000444 R      03
READDATA                 00000244 R      03
READ_BLOCK             = 00000006
READ_STATUS            = 00000004
RESETXFR                 00000435 R      03
RESET_STATUS_ONLY        0000045C R      03
RETREG                   00000389 R      03
RETRYERR                 00000252 R      03
RETURNED_STATUS        = 00000004
SEEK                     0000022A R      03
SIZ...                 = 00000008
SPECOND                  00000414 R      03
SS$_CTRLERR            = 00000054
SS$_DRVERR             = 0000008C
SS$_IVBUFLEN           = 0000034C
SS$_MEDOFL             = 000001A4
SS$_NORMAL             = 00000001
SS$_PARITY             = 000001F4
SS$_TIMEOUT            = 0000022C
SS$_VOLINV             = 00000254
SS$_WRITLCK            = 0000025C
STATUS_RESET           = 00000002
STXCS_A_IE             = 00000040
STXCS_S_ADDRS          = 00000010
STXCS_S_STS            = 00000008
STXCS_V_ADDRS          = 00000008
STXCS_V_RDY            = 00000007
STXCS_V_STS            = 00000018
TRANS_ABORTED          = 00000003
TRANS_COMPLETE         = 00000001
TRANS_CONTINUE         = 00000002
UCB$B_CV_STATE           000000DC
UCB$B_CV_STS             000000DD
UCB$B_DEVCLASS         = 00000040
UCB$B_DEVTYPE          = 00000041
UCB$B_DIPL             = 0000005E
UCB$B_ERTCNT           = 00000080
UCB$B_ERTMAX           = 00000081
```

```
UCB$B_FEX                       = 00000092
UCB$B_FIPL                      = 0000000B
UCB$B_SECTORS                   = 00000044
UCB$B_TRACKS                    = 00000045
UCB$K_CV_LEN                      000000FC
UCB$K_LCL_DISK_LENGTH           = 000000CC
UCB$L_CRB                       = 00000024
UCB$L_CV_ABPC                     000000F8
UCB$L_CV_BUFWIN                   000000E8
UCB$L_CV_CS                       000000CC
UCB$L_CV_IBUF                     000000E0
UCB$L_CV_LBN                      000000F4
UCB$L_CV_MP                       000000D0
UCB$L_CV_MVRTN                    000000E4
UCB$L_DEVCHAR                   = 00000038
UCB$L_DPC                       = 0000009C
UCB$L_FPC                       = 0000000C
UCB$L_IRP                       = 00000058
UCB$L_MAXBLOCK                  = 000000B0
UCB$L_MEDIA                     = 000000BC
UCB$L_MEDIA_ID                  = 0000008C
UCB$L_SVAPTE                    = 00000078
UCB$M_DIAGBUF                   = 00000002
UCB$M_INT                       = 00000002
UCB$M_ONLINE                    = 00000010
UCB$M_POWER                     = 00000020
UCB$M_TIMOUT                    = 00000040
UCB$M_VALID                     = 00000800
UCB$Q_CV_BDAT                     000000EC
UCB$Q_CV_CSMP                     000000D4
UCB$V_INT                       = 00000001
UCB$V_POWER                     = 00000005
UCB$V_VALID                     = 0000000B
UCB$W_BCNT                      = 0000007E
UCB$W_BCR                       = 000000C0
UCB$W_CV_BBC                      000000DE
UCB$W_CYLINDERS                 = 00000046
UCB$W_DEVBUFSIZ                 = 00000042
UCB$W_DEVSTS                    = 00000068
UCB$W_FUNC                      = 0000009A
UCB$W_STS                       = 00000064
UNLOAD                            0000023C R       03
VEC$L_IDB                       = 00000008
VEC$L_INITIAL                   = 0000000C
VEC$L_UNITINIT                  = 00000018
WRITECHECK                        0000022A R       03
WRITEDATA                         00000249 R       03
WRITE_BLOCK                     = 00000005
XFER                              00000324 R       03
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	000000FC (252.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$$$105_PROLOGUE	0000007C (124.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
$$$115_DRIVER	0000079D (1949.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	34	00:00:00.04	00:00:01.74
Command processing	118	00:00:00.42	00:00:05.63
Pass 1	549	00:00:17.57	00:01:13.91
Symbol table sort	0	00:00:02.41	00:00:09.60
Pass 2	238	00:00:03.73	00:00:12.75
Symbol table output	33	00:00:00.20	00:00:00.33
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	977	00:00:24.40	00:01:44.13

The working set limit was 2100 pages.
140986 bytes (276 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2192 non-local and 53 local symbols.
1303 source lines were read in Pass 1, producing 21 object records in Pass 2.
45 pages of virtual memory were used to define 42 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	28
_255DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	38

2392 GETS were required to define 38 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:CVDRIVER/OBJ=OBJ$:CVDRIVER MSRC$:CVDRIVER/UPDATE=(ENH$:CVDRIVER)+EXECML$/LIB

DBDRIVER
LIS

CRDRIVER
LIS

CVDRIVER
LIS

DDDRIVER
LIS

DDDMP
LIS